16th Innovations in Theoretical Computer Science Conference

ITCS 2025, January 7–10, 2025, Columbia University, New York, NY, USA

Edited by

Raghu Meka



Editors

Raghu Meka
University of California, Los Angeles, CA, USA
raghum@cs.ucla.edu

ACM Classification 2012
Mathematics of computing; Theory of computation

ISBN 978-3-95977-361-4

Published online and open access by
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at https://www.dagstuhl.de/dagpub/978-3-95977-361-4.

Publication date
February, 2025

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at https://portal.dnb.de.

License
This work is licensed under a Creative Commons Attribution 4.0 International license (CC-BY 4.0): https://creativecommons.org/licenses/by/4.0/legalcode.
In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:
- Attribution: The work must be attributed to its authors.

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/LIPIcs.ITCS.2025.0

ISBN 978-3-95977-361-4 **ISSN 1868-8969** **https://www.dagstuhl.de/lipics**

LIPIcs – Leibniz International Proceedings in Informatics

LIPIcs is a series of high-quality conference proceedings across all fields in informatics. LIPIcs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

Editorial Board

- Luca Aceto (Reykjavik University, IS and Gran Sasso Science Institute, IT)
- Christel Baier (TU Dresden, DE)
- Roberto Di Cosmo (Inria and Université Paris Cité, FR)
- Faith Ellen (University of Toronto, CA)
- Javier Esparza (TU München, DE)
- Daniel Král' (Masaryk University, Brno, CZ)
- Meena Mahajan (*Chair*, Institute of Mathematical Sciences, Chennai, IN)
- Anca Muscholl (University of Bordeaux, FR)
- Chih-Hao Luke Ong (Nanyang Technological University, SG)
- Phillip Rogaway (University of California, Davis, US)
- Eva Rotenberg (Technical University of Denmark, Lyngby, DK)
- Raimund Seidel (Universität des Saarlandes, Saarbrücken, DE and Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Wadern, DE)
- Pierre Senellart (ENS, Université PSL, Paris, FR)

ISSN 1868-8969

https://www.dagstuhl.de/lipics

Contents

Preface
Raghu Meka 0:xi

List of Authors
..... 0:xiii

Papers

Simple Is COOL: Graded Dispersal and Its Applications for Byzantine Fault Tolerance
Ittai Abraham, Gilad Asharov, and Anirudh Chandramouli 1:1–1:20

Improved Lower Bounds for 3-Query Matching Vector Codes
Divesh Aggarwal, Pranjal Dutta, Zeyong Li, Maciej Obremski, and Sidhant Saraogi 2:1–2:19

Sparsity Lower Bounds for Probabilistic Polynomials
Josh Alman, Arkadev Chattopadhyay, and Ryan Williams 3:1–3:25

A Bicriterion Concentration Inequality and Prophet Inequalities for k-Fold Matroid Unions
Noga Alon, Nick Gravin, Tristan Pollner, Aviad Rubinstein, Hongao Wang, S. Matthew Weinberg, and Qianfan Zhang 4:1–4:22

Edge-Minimum Walk of Modular Length in Polynomial Time
Antoine Amarilli, Benoît Groz, and Nicole Wein 5:1–5:23

Doubly Sub-Linear Interactive Proofs of Proximity
Noga Amir, Oded Goldreich, and Guy N. Rothblum 6:1–6:25

Simultaneous Haar Indistinguishability with Applications to Unclonable Cryptography
Prabhanjan Ananth, Fatih Kaleoglu, and Henry Yuen 7:1–7:23

Single-Round Proofs of Quantumness from Knowledge Assumptions
Petia Arabadjieva, Alexandru Gheorghiu, Victor Gitton, and Tony Metger 8:1–8:16

The Local Hamiltonian Problem for Quasi-Quantum States: A Toy Model for the Quantum PCP Conjecture
Itai Arad and Miklos Santha 9:1–9:1

Algorithmic Collusion Without Threats
Eshwar Ram Arunachaleswaran, Natalie Collina, Sampath Kannan, Aaron Roth, and Juba Ziani 10:1–10:21

Rank Lower Bounds on Non-Local Quantum Computation
Vahid R. Asadi, Eric Culf, and Alex May 11:1–11:18

Stable Matching with Interviews
Itai Ashlagi, Jiale Chen, Mohammad Roghani, and Amin Saberi 12:1–12:19

16th Innovations in Theoretical Computer Science Conference (ITCS 2025).
Editor: Raghu Meka
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Low Sensitivity Hopsets
Vikrant Ashvinkumar, Aaron Bernstein, Chengyuan Deng, Jie Gao, and Nicole Wein 13:1–13:23

The More the Merrier! On Total Coding and Lattice Problems and the Complexity of Finding Multicollisions
Huck Bennett, Surendra Ghentiyala, and Noah Stephens-Davidowitz 14:1–14:22

Exponential-Time Approximation (Schemes) for Vertex-Ordering Problems
Matthias Bentert, Fedor V. Fomin, Tanmay Inamdar, and Saket Saurabh 15:1–15:18

On Fault Tolerant Single-Shot Logical State Preparation and Robust Long-Range Entanglement
Thiago Bergamaschi and Yunchao Liu 16:1–16:9

Random Restrictions of Bounded Low Degree Polynomials Are Juntas
Sreejata Kishor Bhattacharya 17:1–17:21

Finite Matrix Multiplication Algorithms from Infinite Groups
Jonah Blasiak, Henry Cohn, Joshua A. Grochow, Kevin Pratt, and Chris Umans 18:1–18:23

Differential Privacy and Sublinear Time Are Incompatible Sometimes
Jeremiah Blocki, Hendrik Fichtenberger, Elena Grigorescu, and Tamalika Mukherjee 19:1–19:18

Estimating Euclidean Distance to Linearity
Andrej Bogdanov and Lorenzo Taschin 20:1–20:18

Adjacency Labeling Schemes for Small Classes
Édouard Bonnet, Julien Duron, John Sylvester, and Viktor Zamaraev 21:1–21:22

Round-Vs-Resilience Tradeoffs for Binary Feedback Channels
Mark Braverman, Klim Efremenko, Gillat Kol, Raghuvansh R. Saxena, and Zhijun Zhang 22:1–22:23

Accumulation Without Homomorphism
Benedikt Bünz, Pratyush Mishra, Wilson Nguyen, and William Wang 23:1–23:25

Sampling List Packings
Evan Camrud, Ewan Davies, Alex Karduna, and Holden Lee 24:1–24:15

Locally Private Histograms in All Privacy Regimes
Clément L. Canonne and Abigail Gentle 25:1–25:24

Settling the Complexity of Testing Grainedness of Distributions, and Application to Uniformity Testing in the Huge Object Model
Clément L. Canonne, Sayantan Sen, and Joy Qiping Yang 26:1–26:19

The Randomness Complexity of Differential Privacy
Clément L. Canonne, Francis E. Su, and Salil P. Vadhan 27:1–27:21

Coresets for 1-Center in ℓ_1 Metrics
Amir Carmel, Chengzhi Guo, Shaofeng H.-C. Jiang, and Robert Krauthgamer 28:1–28:20

Extracting Dual Solutions via Primal Optimizers
Yair Carmon, Arun Jambulapati, Liam O'Carroll, and Aaron Sidford 29:1–29:24

Provability of the Circuit Size Hierarchy and Its Consequences
Marco Carmosino, Valentine Kabanets, Antonina Kolokolova, Igor C. Oliveira, and Dimitrios Tsintsilidas 30:1–30:22

Quantum Advantage and Lower Bounds in Parallel Query Complexity
Joseph Carolan, Amin Shiraz Gilani, and Mahathi Vempati 31:1–31:14

Succinct Fermion Data Structures
Joseph Carolan and Luke Schaeffer 32:1–32:21

Bosonic Quantum Computational Complexity
Ulysse Chabaud, Michael Joseph, Saeed Mehraban, and Arsalan Motamedi 33:1–33:19

Directed Hypercube Routing, a Generalized Lehman-Ron Theorem, and Monotonicity Testing
Deeparnab Chakrabarty and C. Seshadhri 34:1–34:15

Unraveling Universally Closest Refinements via Symmetric Density Decomposition and Fisher Market Equilibrium
T-H. Hubert Chan and Quan Xue 35:1–35:23

Entry-Specific Matrix Estimation Under Arbitrary Sampling Patterns Through the Lens of Network Flows
Yudong Chen, Xumei Xi, and Christina Lee Yu 36:1–36:2

A Lower Bound on the Trace Norm of Boolean Matrices and Its Applications
Tsun-Ming Cheung, Hamed Hatami, Kaave Hosseini, Aleksandar Nikolov, Toniann Pitassi, and Morgan Shirley 37:1–37:15

Backdoor Defense, Learnability and Obfuscation
Paul Christiano, Jacob Hilton, Victor Lecomte, and Mark Xu 38:1–38:21

Data Reconstruction: When You See It and When You Don't
Edith Cohen, Haim Kaplan, Yishay Mansour, Shay Moran, Kobbi Nissim, Uri Stemmer, and Eliad Tsfadia 39:1–39:23

Derandomized Squaring: An Analytical Insight into Its True Behavior
Gil Cohen, Itay Cohen, Gal Maor, and Yuval Peled 40:1–40:24

Query Complexity of Stochastic Minimum Vertex Cover
Mahsa Derakhshan, Mohammad Saneian, and Zhiyang Xun 41:1–41:12

Nearest Neighbor Complexity and Boolean Circuits
Mason DiCicco, Vladimir Podolskii, and Daniel Reichman 42:1–42:23

Space Complexity of Minimum Cut Problems in Single-Pass Streams
Matthew Ding, Alexandro Garces, Jason Li, Honghao Lin, Jelani Nelson, Vihan Shah, and David P. Woodruff 43:1–43:23

Learning-Augmented Streaming Algorithms for Approximating MAX-CUT
Yinhao Dong, Pan Peng, and Ali Vakilian 44:1–44:24

Distributed and Parallel Low-Diameter Decompositions for Arbitrary and Restricted Graphs
Jinfeng Dou, Thorsten Götte, Henning Hillebrandt, Christian Scheideler, and Julian Werthmann 45:1–45:26

Data-Driven Solution Portfolios
Marina Drygala, Silvio Lattanzi, Andreas Maggiori, Miltiadis Stouras, Ola Svensson, and Sergei Vassilvitskii 46:1–46:15

Concentration of Submodular Functions and Read-k Families Under Negative Dependence
Sharmila Duppala, George Z. Li, Juan Luque, Aravind Srinivasan, and Renata Valieva 47:1–47:16

Robust Restaking Networks
Naveen Durvasula and Tim Roughgarden 48:1–48:21

Confusion Matrix Design for Downstream Decision-Making
Yiding Feng and Wei Tang 49:1–49:1

Fully Characterizing Lossy Catalytic Computation
Marten Folkertsma, Ian Mertz, Florian Speelman, and Quinten Tupker 50:1–50:13

Parameterized Geometric Graph Modification with Disk Scaling
Fedor V. Fomin, Petr A. Golovach, Tanmay Inamdar, Saket Saurabh, and Meirav Zehavi 51:1–51:17

Combinatorial Pen Testing (Or Consumer Surplus of Deferred-Acceptance Auctions)
Aadityan Ganesh and Jason Hartline 52:1–52:22

Differential Privacy on Trust Graphs
Badih Ghazi, Ravi Kumar, Pasin Manurangsi, and Serena Wang 53:1–53:23

Partial Minimum Branching Program Size Problem Is ETH-Hard
Ludmila Glinskih and Artur Riazanov 54:1–54:22

Completeness Theorems for k-SUM and Geometric Friends: Deciding Fragments of Linear Integer Arithmetic
Geri Gokaj and Marvin Künnemann 55:1–55:25

Incompressible Functional Encryption
Rishab Goyal, Venkata Koppula, Mahesh Sreekumar Rajasree, and Aman Verma 56:1–56:22

Deterministic Approximation for the Volume of the Truncated Fractional Matching Polytope
Heng Guo and Vishvajeet N 57:1–57:14

The Computational Complexity of Factored Graphs
Shreya Gupta, Boyang Huang, Russell Impagliazzo, Stanley Woo, and Christopher Ye 58:1–58:19

Error Correction for Message Streams
Meghal Gupta and Rachel Yun Zhang 59:1–59:18

List Decoding Bounds for Binary Codes with Noiseless Feedback
Meghal Gupta and Rachel Yun Zhang 60:1–60:20

Hardness of Sampling for the Anti-Ferromagnetic Ising Model on Random Graphs
Neng Huang, Will Perkins, and Aaron Potechin 61:1–61:23

A High Dimensional Cramer's Rule Connecting Homogeneous Multilinear Equations to Hyperdeterminants
Antoine Joux and Anand Kumar Narayanan 62:1–62:16

Complexity Classification of Product State Problems for Local Hamiltonians
John Kallaugher, Ojas Parekh, Kevin Thompson, Yipu Wang, and Justin Yirka 63:1–63:32

A Universal Sequence of Tensors for the Asymptotic Rank Conjecture
Petteri Kaski and Mateusz Michałek 64:1–64:24

Online Versus Offline Adversaries in Property Testing
Esty Kelman, Ephraim Linder, and Sofya Raskhodnikova 65:1–65:18

Graph Reconstruction via MIS Queries
Christian Konrad, Conor O'Sullivan, and Victor Traistaru 66:1–66:19

Error-Correcting Graph Codes
Swastik Kopparty, Aditya Potukuchi, and Harry Sha 67:1–67:20

New Pseudorandom Generators and Correlation Bounds Using Extractors
Vinayak M. Kumar 68:1–68:23

Approximate Unitary k-Designs from Shallow, Low-Communication Circuits
Nicholas LaRacuente and Felix Leditzky 69:1–69:2

Facility Location on High-Dimensional Euclidean Spaces
Euiwoong Lee and Kijun Shin 70:1–70:21

Toward the Impossibility of Perfect Complete Quantum PKE from OWFs
Longcheng Li, Qian Li, Xingjian Li, and Qipeng Liu 71:1–71:16

Information Design with Unknown Prior
Tao Lin and Ce Li 72:1–72:1

On White-Box Learning and Public-Key Encryption
Yanyi Liu, Noam Mazor, and Rafael Pass 73:1–73:22

Sublinear Metric Steiner Tree via Improved Bounds for Set Cover
Sepideh Mahabadi, Mohammad Roghani, Jakub Tarnawski, and Ali Vakilian 74:1–74:24

Gadgetless Lifting Beats Round Elimination: Improved Lower Bounds for Pointer Chasing
Xinyu Mao, Guangxu Yang, and Jiapeng Zhang 75:1–75:14

A Quantum Unique Games Conjecture
Hamoon Mousavi and Taro Spirig 76:1–76:16

Sketching, Moment Estimation, and the Lévy-Khintchine Representation Theorem
Seth Pettie and Dingyu Wang 77:1–77:23

Randomized Lifting to Semi-Structured Communication Complexity via Linear Diversity
Vladimir Podolskii and Alexander Shekhovtsov 78:1–78:21

Catalytic Communication
Edward Pyne, Nathan S. Sheffield, and William Wang 79:1–79:24

Diversity in Evolutionary Dynamics
Yuval Rabani, Leonard J. Schulman, and Alistair Sinclair 80:1–80:3

Online Balanced Allocation of Dynamic Components
Rajmohan Rajaraman and Omer Wasim 81:1–81:23

Tight Bounds on List-Decodable and List-Recoverable Zero-Rate Codes
Nicolas Resch, Chen Yuan, and Yihan Zhang 82:1–82:21

Fine-Grained Equivalence for Problems Related to Integer Linear Programming
Lars Rohwedder and Karol Węgrzycki 83:1–83:18

Quantum Communication Complexity of Classical Auctions
Aviad Rubinstein and Zixin Zhou 84:1–84:27

Quantum 2-SAT on Low Dimensional Systems Is QMA_1-Complete: Direct Embeddings and Black-Box Simulation
Dorian Rudolph, Sevag Gharibian, and Daniel Nagaj 85:1–85:24

Eulerian Orientations and Hadamard Codes: A Novel Connection via Counting
Shuai Shao and Zhuxiao Tang 86:1–86:13

When to Give up on a Parallel Implementation
Nathan S. Sheffield and Alek Westover 87:1–87:18

Detecting and Correcting Computationally Bounded Errors: A Simple Construction Under Minimal Assumptions
Jad Silbak and Daniel Wichs 88:1–88:23

Optimal Communication Complexity of Chained Index
Janani Sundaresan 89:1–89:18

Polynomial Size, Short-Circuit Resilient Circuits for NC
Yael Tauman Kalai and Raghuvansh R. Saxena 90:1–90:18

Simple Norm Bounds for Polynomial Random Matrices via Decoupling
Madhur Tulsiani and June Wu 91:1–91:22

Listing 6-Cycles in Sparse Graphs
Virginia Vassilevska Williams and Alek Westover 92:1–92:21

Polynomials, Divided Differences, and Codes
S. Venkitesh 93:1–93:25

New Direct Sum Tests
Alek Westover, Edward Yu, and Kai Zhe Zheng 94:1–94:26

Toward Separating QMA from QCMA with a Classical Oracle
Mark Zhandry 95:1–95:19

Formulations and Constructions of Remote State Preparation with Verifiability, with Applications
Jiayu Zhang 96:1–96:19

Preface

The papers in this volume were presented at the 15th Innovations in Theoretical Computer Science (ITCS 2025) conference. The conference was held from January 7 to January 10, 2025 at Columbia University

ITCS seeks to promote research with innovative or bold agendas, which could be conceptual, technical, or methodological, and whose message will advance and inspire the greater theory community. Some examples of the kind of papers that the conference aims to feature are those introducing a new concept, model or understanding; opening a new line of inquiry within traditional or interdisciplinary areas; introducing new mathematical techniques and methodologies, or new applications of known techniques; putting forth a bold, even if preliminary, vision or line of attack; making interesting progress on traditional research directions; or unearthing novel or surprising connections between different topics.

The conference received 303 submissions of which the program committee accepted 96 papers. The submission pool was strong, which explains the high acceptance rate. The conference format was single-session with the goal of promoting a sense of community, and promoting the exchange of ideas between different areas of theoretical computer science. Given this and the sizable number of accepted papers, each talk at the conference was only about 10 minutes long. The authors of each paper were thus requested to submit a 20–25 minute video, which are linked from the conference website. Among the many strong submissions, two papers *Random Restrictions of Bounded Low Degree Polynomials Are Juntas*, by Sreejata Kishor Bhattacharya (Tata Institute of Fundamental Research) and *Error Correction for Message Streams*, by Rachel Yun Zhang (MIT); Meghal Gupta (UC Berkeley) received the Best Student Paper Award.

The program committee (PC) consisted of 37 fantastic members (excluding the chair): Ittai Abraham (Intel), Sepehr Assadi (University of Waterloo), Andris Ambainis (University of Latvia), Prabhanjan Ananth (University of California, Santa Barbara), Srinivasan Arunachalam (IBM Research), Hadley Black (University of California, San Diego), Adam Bouland (Stanford University), Eshan Chattopadhyay (Cornell University), Lijie Chen (University of California, Berkeley), Rishabh Goyal (University of Wisconsin-Madison), Prahladh Harsha (Tata Institute of Fundamental Research), Aayush Jain (Carnegie Mellon University), Rahul Jain (National University of Singapore), Michael Kapralov (École polytechnique fédérale de Lausanne), Robert Kleinberg (Cornell University), Vasilis Kontonis (University of Texas at Austin), Pravesh Kothari (Princeton University), Ravi Kumar (Google Research), Jason Li (Carnegie Mellon University), Daniel Lokshtanov (University of California, Santa Barbara), Pasin Manurangasi (Google Research), Dor Minzer (Massachussetts Institute of Technology), Chinmay Nikhre (University of Washington, Seattle), Shayan Oveis Gharan (University of Washington, Seattle), Prasad Raghavendra (University of California, Berkeley), Guy Rothblum (Apple), Shubhangi Saraf (University of Toronto), Sushant Sachdeva (University of Toronto), Tselil Schramm (Stanford University), Sahil Singla (Georgia Institute of Technology), Srikanth Srinivasan (University of Copenhagen), Avishay Tal (University of California, Berkeley), Roei Tell (University of Toronto), Christos Tzamos (University of Athens), David Wajc (Technion), Ryan Williams (Massachussetts Institute of Technology), Jiapeng Zhang (University of Southern California).

I am extremely grateful to all members of the PC, who worked really hard under very tight time constraints to produce a fantastic program as well as provide useful feedback to the authors, of accepted and rejected papers alike. The review process was double-blind, and

16th Innovations in Theoretical Computer Science Conference (ITCS 2025).
Editor: Raghu Meka
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

throughout the review process the PC members did not have access to the identities of the authors on the hotcrp conference review software. The rationale behind the double-blind process was to help PC members and external reviewers come to a judgment about the paper without unconscious bias, but it was not intended to make it impossible for them to discover who the authors are. As such, the authors were free to post their papers or otherwise make them publicly available. In addition, while the conference did not have a formal rebuttal process, the PC engaged in interactive communication with the authors throughout the review process, coordinated by the chair, to address any misunderstandings (e.g., concerns about errors or missing attributions).

The program schedule was divided into sessions, each consisting of 6–7 papers. Following ITCS tradition, the chair of each session was tasked with "ranting" about the papers in the session, emphasizing their contributions and the ways in which they are innovative, and when possible tying all the papers in the session together. The program also continued with the wonderful ITCS tradition of "graduating bits," where students and postdocs looking for academic jobs or postdoc opportunities give a very short (few minutes) presentation. In addition, the local chairs also organized a fun game night during the conference.

I would like to thank Columbia University for hosting the event. Local chairs Josh Alman and Alexander Andoni were outstanding in their organization, smoothly managing the many unexpected issues that arose during the conference. I am also grateful to ITCS Steering Committee chair Guy Rothblum for his guidance, for patiently answering all my questions, and running the event smoothly. Josh, Alexander, and Guy were tremendously helpful in shouldering extra responsibilities during the physical conference, as I was unable to attend due to the emergency situation in Los Angeles at the time. Their efforts were essential to the conference's success.

Raghu Meka
Department of Computer Science,
University of California, Los Angeles, CA

List of Authors

Ittai Abraham (1)
Intel Labs, Petah Tikva, Israel

Divesh Aggarwal (2)
Centre for Quantum Technologies,
National University of Singapore, Singapore

Josh Alman (3)
Columbia University, New York, NY, USA

Noga Alon (4)
Department of Mathematics, Princeton University, NJ, USA;
Schools of Mathematics and Computer Science, Tel Aviv University, Tel Aviv, Israel

Antoine Amarilli (5)
Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France;
LTCI, Télécom Paris, Institut polytechnique de Paris, France

Noga Amir (6)
Weizmann Institute of Science, Rehovot, Israel

Prabhanjan Ananth (7)
University of California, Santa Barbara, CA, USA

Petia Arabadjieva (8)
Institute for Theoretical Physics,
ETH Zurich, Switzerland

Itai Arad (9)
Centre for Quantum Technologies, Singapore

Eshwar Ram Arunachaleswaran (10)
Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA

Vahid R. Asadi (11)
University of Waterloo, Canada

Gilad Asharov (1)
Bar-Ilan University, Ramat-Gan, Israel

Itai Ashlagi (12)
Stanford University, CA, USA

Vikrant Ashvinkumar (13)
Rutgers University, Piscataway, NJ, USA

Huck Bennett (14)
University of Colorado Boulder, CO, USA

Matthias Bentert (15)
University of Bergen, Norway

Thiago Bergamaschi (16)
University of California Berkeley, CA, USA

Aaron Bernstein (13)
New York University, NY, USA

Sreejata Kishor Bhattacharya (17)
School of Technology and Computer Science, Tata Institute of Fundamental Research, Mumbai, India

Jonah Blasiak (18)
Department of Mathematics, Drexel University, Philadelphia, PA, USA

Jeremiah Blocki (19)
Purdue University, West Lafayette, IN, USA

Andrej Bogdanov (20)
University of Ottawa, Canada

Édouard Bonnet (21)
Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France

Mark Braverman (22)
Princeton University, NJ, USA

Benedikt Bünz (23)
New York University, NY, USA

Igor C. Oliveira (30)
University of Warwick, UK

Evan Camrud (24)
Department of Mathematics and Statistics, Middlebury College, VT, USA

Clément L. Canonne (25, 26, 27)
School of Computer Science,
University of Sydney, Australia

Amir Carmel (28)
Weizmann Institute of Science, Rehovot, Israel

Yair Carmon (29)
Tel Aviv University, Israel

Marco Carmosino (30)
IBM Research, Cambridge, MA, USA

Joseph Carolan (31, 32)
University of Maryland, College Park, MD, USA

Ulysse Chabaud (33)
DIENS, ENS, PSL University, CNRS, INRIA, Paris, France

16th Innovations in Theoretical Computer Science Conference (ITCS 2025).
Editor: Raghu Meka
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Deeparnab Chakrabarty (34)
Dartmouth College, Hanover, NH, USA

T-H. Hubert Chan (35)
Department of Computer Science,
University of Hong Kong, Hong Kong

Anirudh Chandramouli (1)
Bar-Ilan University, Ramat-Gan, Israel

Arkadev Chattopadhyay (3)
TIFR Mumbai, India

Jiale Chen (12)
Stanford University, CA, USA

Yudong Chen (36)
Department of Computer Sciences,
University of Wisconsin-Madison, WI, USA

Tsun-Ming Cheung (37)
School of Computer Science, McGill University,
Montreal, Canada

Paul Christiano (38)
Alignment Research Center, Berkeley, CA, USA

Edith Cohen (39)
Google Research, Mountain View, CA, USA;
Tel Aviv University, Israel

Gil Cohen (40)
Tel Aviv University, Israel

Itay Cohen (40)
Tel Aviv University, Israel

Henry Cohn (18)
Microsoft Research New England,
One Memorial Drive, Cambridge, MA, USA

Natalie Collina (10)
Department of Computer and Information
Science, University of Pennsylvania,
Philadelphia, PA, USA

Eric Culf (11)
University of Waterloo, Canada

Ewan Davies (24)
Department of Computer Science, Colorado
State University, Fort Collins, CO, USA

Chengyuan Deng (13)
Rutgers University, Piscataway, NJ, USA

Mahsa Derakhshan (41)
Northeastern University, Boston, MA, USA

Mason DiCicco (42)
Worcester Polytechnic Institute, MA, USA

Matthew Ding (43)
University of California, Berkeley, CA, USA

Yinhao Dong (44)
School of Computer Science and Technology,
University of Science and Technology of China,
Hefei, China

Jinfeng Dou (45)
Paderborn University, Germany

Marina Drygala (46)
EPFL, Lausanne, Switzerland

Sharmila Duppala (47)
University of Maryland, College Park, MD, USA

Julien Duron (21)
Univ Lyon, CNRS, ENS de Lyon, Université
Claude Bernard Lyon 1, LIP UMR5668, France

Naveen Durvasula (48)
Columbia University, New York, NY, USA

Pranjal Dutta (2)
National University of Singapore, Singapore

Klim Efremenko (22)
Ben-Gurion University, Beer-Sheva, Israel

Yiding Feng (49)
Department of Industrial Engineering and
Decision Analytics, Hong Kong University of
Science and Technology, Hong Kong

Hendrik Fichtenberger (19)
Google Research, Zürich, Switzerland

Marten Folkertsma (50)
CWI, Amsterdam, The Netherlands;
QuSoft, Amsterdam, The Netherlands

Fedor V. Fomin (15, 51)
University of Bergen, Norway

Aadityan Ganesh (52)
Princeton University, NJ, USA

Jie Gao (13)
Rutgers University, Piscataway, NJ, USA

Alexandro Garces (43)
Massachusetts Institute of Technology,
Cambridge, MA, USA

Abigail Gentle (25)
School of Computer Science,
University of Sydney, Australia

Sevag Gharibian (85)
Department of Computer Science and Institute for Photonic Quantum Systems (PhoQS), Paderborn University, Germany

Badih Ghazi (53)
Google Research, Mountain View, CA, USA

Surendra Ghentiyala (14)
Cornell University, Ithaca, NY, USA

Alexandru Gheorghiu (8)
Department of Computer Science and Engineering, Chalmers University of Technology, Göteborg, Sweden

Amin Shiraz Gilani (31)
University of Maryland, College Park, MD, USA

Victor Gitton (8)
Institute for Theoretical Physics, ETH Zurich, Switzerland

Ludmila Glinskih (54)
Google, Mountain View, CA, USA

Geri Gokaj (55)
Karlsruhe Institute of Technology, Germany

Oded Goldreich (6)
Weizmann Institute of Science, Rehovot, Israel

Petr A. Golovach (51)
University of Bergen, Norway

Rishab Goyal (56)
University of Wisconsin-Madison, WI, USA

Nick Gravin (4)
Key Laboratory of Interdisciplinary Research of Computation and Economics, Shanghai University of Finance and Economics, China

Elena Grigorescu (19)
University of Waterloo, ON, Canada

Joshua A. Grochow (18)
Departments of Computer Science and Mathematics, University of Colorado Boulder, CO, USA

Benoît Groz (5)
Paris-Saclay University, CNRS, LISN, France

Chengzhi Guo (28)
Peking University, China

Heng Guo (57)
School of Informatics, University of Edinburgh, UK

Meghal Gupta (59, 60)
University of California Berkeley, CA, USA

Shreya Gupta (58)
University of California San Diego, La Jolla, CA, USA

Thorsten Götte (45)
University of Hamburg, Germany

Jason Hartline (52)
Northwestern University, Evanston, IL, USA

Hamed Hatami (37)
School of Computer Science, McGill University, Montreal, Canada

Henning Hillebrandt (45)
Paderborn University, Germany

Jacob Hilton (38)
Alignment Research Center, Berkeley, CA, USA

Kaave Hosseini (37)
Department of Computer Science, University of Rochester, NY, USA

Boyang Huang (58)
University of California San Diego, La Jolla, CA, USA

Neng Huang (61)
University of Michigan, Ann Arbor, MI, USA

Russell Impagliazzo (58)
University of California San Diego, La Jolla, CA, USA

Tanmay Inamdar (15, 51)
Indian Institute of Technology Jodhpur, India

Arun Jambulapati (29)
University of Michigan, Ann Arbor, MI, USA

Shaofeng H.-C. Jiang (28)
Peking University, China

Michael Joseph (33)
Department of Computer Science, Tufts University, Medford, MA, USA

Antoine Joux (62)
CISPA – Helmholtz Center for Information Security, Saarbrücken, Germany

Valentine Kabanets (30)
Simon Fraser University, Burnaby, Canada

Fatih Kaleoglu (7)
University of California, Santa Barbara, CA, USA

John Kallaugher (63)
Sandia National Laboratories, Albuquerque, NM, USA

Sampath Kannan (10)
Simons Institute for the Theory of Computing, University of California, Berkeley, CA, USA

Haim Kaplan (39)
Tel Aviv University, Israel; Google Research, Tel Aviv, Israel

Alex Karduna (24)
Department of Computer Science, Colorado State University, Fort Collins, CO, USA

Petteri Kaski (64)
Department of Computer Science, Aalto University, Finland

Esty Kelman (65)
Boston University, MA, USA; Massachusetts Institute of Technology, Cambridge, MA, USA

Gillat Kol (22)
Princeton University, NJ, USA

Antonina Kolokolova (30)
Memorial University of Newfoundland, St. John, Canada

Christian Konrad (66)
School of Computer Science, University of Bristol, UK

Swastik Kopparty (67)
Department of Mathematics and Department of Computer Science, University of Toronto, Canada

Venkata Koppula (56)
IIT Delhi, India

Robert Krauthgamer (28)
Weizmann Institute of Science, Rehovot, Israel

Ravi Kumar (53)
Google Research, Mountain View, CA, USA

Vinayak M. Kumar (68)
University of Texas at Austin, TX, USA

Marvin Künnemann (55)
Karlsruhe Institute of Technology, Germany

Nicholas LaRacuente (69)
Department of Computer Science, University of Indiana Bloomington, IN, USA

Silvio Lattanzi (46)
Google Research, Barecelona, Spain

Victor Lecomte (38)
Alignment Research Center, Berkeley, CA, USA

Felix Leditzky (69)
Department of Mathematics, University of Illinois Urbana-Champaign, IL, USA; Illinois Quantum Information Science and Technology Center, Urbana, IL, USA

Euiwoong Lee (70)
University of Michigan, Ann Arbor, MI, USA

Holden Lee (24)
Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD, USA

Ce Li (72)
Department of Economics, Boston University, MA, USA

George Z. Li (47)
Carnegie Mellon University, Pittsburgh, PA, USA

Jason Li (43)
Carnegie Mellon University, Pittsburgh, PA, USA

Longcheng Li (71)
State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

Qian Li (71)
Shenzhen International Center for Industrial and Applied Mathematics, Shenzhen Research Institute of Big Data, China

Xingjian Li (71)
Tsinghua University, Beijing, China

Zeyong Li (2)
Centre for Quantum Technologies, National University of Singapore, Singapore

Honghao Lin (43)
Carnegie Mellon University, Pittsburgh, PA, USA

Tao Lin (72)
John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA

Ephraim Linder (65)
Boston University, MA, USA

Qipeng Liu (71)
University of California, San Diego, La Jolla, CA, USA

Yanyi Liu (73)
Cornell Tech, New York, NY, USA

Yunchao Liu (16)
University of California Berkeley, CA, USA;
Harvard University, Cambridge, MA, USA

Juan Luque (47)
University of Maryland, College Park, MD, USA

Andreas Maggiori (46)
Columbia University, New York, NY, USA

Sepideh Mahabadi (74)
Microsoft Research, Redmond, WA, USA

Yishay Mansour (39)
Tel Aviv University, Israel;
Google Research, Tel Aviv, Israel

Pasin Manurangsi (53)
Google Research, Bangkok, Thailand

Xinyu Mao (75)
Thomas Lord Department of Computer Science,
University of Southern California, Los Angeles,
CA, USA

Gal Maor (40)
Tel Aviv University, Israel

Alex May (11)
Perimeter Institute for Theoretical Physics,
Waterloo, Canada;
University of Waterloo, Canada

Noam Mazor (73)
Tel Aviv University, Israel

Saeed Mehraban (33)
Department of Computer Science,
Tufts University, Medford, MA, USA

Ian Mertz (50)
University of Warwick, UK

Tony Metger (8)
Institute for Theoretical Physics,
ETH Zurich, Switzerland

Mateusz Michałek (64)
Fachbereich Mathematik und Statistik,
Universität Konstanz, Germany

Pratyush Mishra (23)
University of Pennsylvania,
Philadelphia, PA, USA

Shay Moran (39)
Technion, Haifa, Israel;
Google Research, Tel Aviv, Israel

Arsalan Motamedi (33)
Institute for Quantum Computing,
Waterloo, Canada

Hamoon Mousavi (76)
Simons Institute for Theoretical Computer
Science, University of California, Berkeley,
CA, USA

Tamalika Mukherjee (19)
Columbia University, New York, NY, USA

Vishvajeet N (57)
School of Informatics,
University of Edinburgh, UK

Daniel Nagaj (85)
Institute of Physics, Slovak Academy of Sciences,
Bratislava, Slovakia

Anand Kumar Narayanan (62)
SandboxAQ, Palo Alto, CA, USA

Jelani Nelson (43)
University of California, Berkeley, CA, USA

Wilson Nguyen (23)
Stanford University, CA, USA

Aleksandar Nikolov (37)
Department of Computer Science,
University of Toronto, Canada

Kobbi Nissim (39)
Georgetown University, Washington, DC, USA;
Work done while at Google Research, Tel Aviv,
Israel

Liam O'Carroll (29)
Stanford University, CA, USA

Conor O'Sullivan (66)
School of Computer Science,
University of Bristol, UK

Maciej Obremski (2)
National University of Singapore, Singapore

Ojas Parekh (63)
Sandia National Laboratories, Albuquerque,
NM, USA

Rafael Pass (73)
Cornell Tech, New York, NY, USA;
Technion, Haifa, Israel;
Tel Aviv University, Israel

Yuval Peled (40)
The Hebrew University of Jerusalem, Israel

Pan Peng (44)
School of Computer Science and Technology, University of Science and Technology of China, Hefei, China

Will Perkins (61)
Georgia Institute of Technology, Atlanta, GA, USA

Seth Pettie (77)
University of Michigan, Ann Arbor, MI, USA

Toniann Pitassi (37)
Department of Computer Science, Columbia University, New York, NY, USA

Vladimir Podolskii (42, 78)
Tufts University, Medford, MA, USA

Tristan Pollner (4)
Department of Management Science and Engineering, Stanford University, CA, USA

Aaron Potechin (61)
University of Chicago, IL, USA

Aditya Potukuchi (67)
Department of Electrical Engineering and Computer Science, York University, Toronto, Canada

Kevin Pratt (18)
Department of Computer Science, Courant Institute of Mathematical Sciences, New York, NY, USA

Edward Pyne (79)
MIT, Cambridge, MA, USA

Yuval Rabani (80)
The Hebrew University of Jerusalem, Israel

Rajmohan Rajaraman (81)
Northeastern University, Boston, MA, USA

Mahesh Sreekumar Rajasree (56)
IIT Delhi, India

Sofya Raskhodnikova (65)
Boston University, MA, USA

Daniel Reichman (42)
Worcester Polytechnic Institute, MA, USA

Nicolas Resch (82)
Informatics' Institute, University of Amsterdam, The Netherlands

Artur Riazanov (54)
EPFL, Lausanne, Switzerland

Mohammad Roghani (12, 74)
Stanford University, CA, USA

Lars Rohwedder (83)
University of Southern Denmark (SDU), Odense, Denmark

Aaron Roth (10)
Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA

Guy N. Rothblum (6)
Apple, Cupertino, CA, USA

Tim Roughgarden (48)
a16zcrypto, New York, NY, USA;
Columbia University, New York, NY, USA

Aviad Rubinstein (4, 84)
Department of Computer Science, Stanford University, CA, USA

Dorian Rudolph (85)
Department of Computer Science and Institute for Photonic Quantum Systems (PhoQS), Paderborn University, Germany

Amin Saberi (12)
Stanford University, CA, USA

Mohammad Saneian (41)
Northeastern University, Boston, MA, USA

Miklos Santha (9)
Centre for Quantum Technologies, Singapore

Sidhant Saraogi (2)
Georgetown University, Washington, DC, USA

Saket Saurabh (15, 51)
The Institute of Mathematical Sciences, Chennai, India;
University of Bergen, Norway

Raghuvansh R. Saxena (22, 90)
Tata Institute of Fundamental Research, Mumbai, India

Luke Schaeffer (32)
University of Waterloo, Canada

Christian Scheideler (45)
Paderborn University, Germany

Leonard J. Schulman (80)
California Institute of Technology, Pasadena, CA, USA

Sayantan Sen (26)
Centre for Quantum Technologies, National University of Singapore, Singapore

C. Seshadhri (34)
University of California, Santa Cruz, CA, USA

Harry Sha (67)
Department of Mathematics and Department of Computer Science, University of Toronto, Canada

Vihan Shah (43)
University of Waterloo, Canada

Shuai Shao (86)
School of Computer Science and Technology & Hefei National Laboratory, University of Science and Technology of China, Hefei, China

Nathan S. Sheffield (79, 87)
MIT, Cambridge, MA, USA

Alexander Shekhovtsov (78)
Moscow Institute of Physics and Technology, Russia

Kijun Shin (70)
Seoul National University, South Korea

Morgan Shirley (37)
Department of Computer Science, University of Victoria, Canada

Aaron Sidford (29)
Stanford University, CA, USA

Jad Silbak (88)
Northeastern University, Boston, MA, USA

Alistair Sinclair (80)
University of California, Berkeley, CA, USA

Florian Speelman (50)
University of Amsterdam, The Netherlands

Taro Spirig (76)
QMATH, Department of Mathematical Sciences, University of Copenhagen, Denmark

Aravind Srinivasan (47)
University of Maryland, College Park, MD, USA

Uri Stemmer (39)
Tel Aviv University, Israel; Google Research, Tel Aviv, Israel

Noah Stephens-Davidowitz (14)
Cornell University, Ithaca, NY, USA

Miltiadis Stouras (46)
EPFL, Lausanne, Switzerland

Francis E. Su (27)
Harvey Mudd College, Claremont, CA, USA

Janani Sundaresan (89)
Cheriton School of Computer Science, University of Waterloo, Canada

Ola Svensson (46)
EPFL, Lausanne, Switzerland

John Sylvester (21)
Department of Computer Science, University of Liverpool, UK

Wei Tang (49)
Department of Decisions, Operations and Technology, Chinese University of Hong Kong, Hong Kong

Zhuxiao Tang (86)
School of the Gifted Young, University of Science and Technology of China, Hefei, China

Jakub Tarnawski (74)
Microsoft Research, Redmond, WA, USA

Lorenzo Taschin (20)
EPFL, Lausanne, Switzerland

Yael Tauman Kalai (90)
Microsoft Research, Cambridge, MA, USA; MIT, Cambridge, MA, USA

Kevin Thompson (63)
Sandia National Laboratories, Albuquerque, NM, USA

Victor Traistaru (66)
School of Computer Science, University of Bristol, UK

Eliad Tsfadia (39)
Georgetown University, Washington, DC, USA

Dimitrios Tsintsilidas (30)
University of Warwick, UK

Madhur Tulsiani (91)
Toyota Technological Institute at Chicago, IL, USA

Quinten Tupker (50)
CWI, Amsterdam, The Netherlands

Chris Umans (18)
Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA

Salil P. Vadhan (27)
Harvard University, Cambridge, MA, USA

Ali Vakilian (44, 74)
Toyota Technological Institute at Chicago (TTIC), IL, USA

Renata Valieva (47)
University of Maryland, College Park, MD, USA

Virginia Vassilevska Williams (92)
MIT, Cambridge, MA, USA

Sergei Vassilvitskii (46)
Google Research, New York, NY, USA

Mahathi Vempati (31)
University of Maryland,
College Park, MD, USA

S. Venkitesh (93)
Tel Aviv University, Israel

Aman Verma (56)
IIT Delhi, India

Dingyu Wang (77)
University of Michigan, Ann Arbor, MI, USA

Hongao Wang (4)
Department of Computer Science, Purdue University, West Lafayette, IN, USA

Serena Wang (53)
Google Research, Mountain View, CA, USA;
Harvard University, Cambridge, MA, USA

William Wang (23, 79)
New York University, NY, USA;
MIT, Cambridge, MA, USA

Yipu Wang (63)
Sandia National Laboratories,
Albuquerque, NM, USA

Omer Wasim (81)
Northeastern University, Boston, MA, USA

Nicole Wein (5, 13)
University of Michigan, Ann Arbor, MI, USA

S. Matthew Weinberg (4)
Department of Computer Science,
Princeton University, NJ, USA

Julian Werthmann (45)
Paderborn University, Germany

Alek Westover (87, 92, 94)
MIT, Cambridge, MA, USA

Daniel Wichs (88)
Northeastern University, Boston, MA, USA;
NTT Research, Sunnyvale, CA, USA

Ryan Williams (3)
CSAIL and EECS, MIT, Cambridge, MA, USA

Stanley Woo (58)
University of California San Diego,
La Jolla, CA, USA

David P. Woodruff (43)
Carnegie Mellon University,
Pittsburgh, PA, USA

June Wu (91)
University of Chicago, IL, USA

Karol Węgrzycki (83)
Saarland University, Saarbrücken, Germany;
Max Planck Institute for Informatics,
Saarbrücken, Germany

Xumei Xi (36)
School of Operations Research and Information Engineering, Cornell University, Ithaca, NY, USA

Mark Xu (38)
Alignment Research Center, Berkeley, CA, USA

Quan Xue (35)
Department of Computer Science,
University of Hong Kong, Hong Kong

Zhiyang Xun (41)
University of Texas at Austin, TX, USA

Guangxu Yang (75)
Thomas Lord Department of Computer Science, University of Southern California, Los Angeles, CA, USA

Joy Qiping Yang (26)
School of Computer Science,
University of Sydney, Australia

Christopher Ye (58)
University of California San Diego,
La Jolla, CA, USA

Justin Yirka (63)
Sandia National Laboratories,
Albuquerque, NM, USA;
The University of Texas at Austin, TX, USA

Christina Lee Yu (36)
School of Operations Research and Information Engineering, Cornell University, Ithaca, NY, USA

Edward Yu (94)
MIT, Cambridge, MA, USA

Chen Yuan (82)
School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China

Henry Yuen (7)
Columbia University, New York, NY, USA

Viktor Zamaraev (21)
Department of Computer Science,
University of Liverpool, UK

Meirav Zehavi (51)
Ben-Gurion University of the Negev,
Beer-Sheva, Israel

Mark Zhandry (95)
NTT Research, Sunnyvale, CA, USA

Jiapeng Zhang (75)
Thomas Lord Department of Computer Science,
University of Southern California, Los Angeles,
CA, USA

Jiayu Zhang (96)
Zhongguancun Laboratory, Beijing, China

Qianfan Zhang (4)
Department of Computer Science,
Princeton University, NJ, USA

Rachel Yun Zhang (59, 60)
MIT, Cambridge, MA, USA

Yihan Zhang (82)
Institute of Science and Technology Austria,
Klosterneuburg, Austria

Zhijun Zhang (22)
Princeton University, NJ, USA

Kai Zhe Zheng (94)
MIT, Cambridge, MA, USA

Zixin Zhou (84)
Department of Computer Science,
Stanford University, CA, USA

Juba Ziani (10)
ISyE, Georgia Tech, Atlanta, GA, USA

Simple Is COOL: Graded Dispersal and Its Applications for Byzantine Fault Tolerance

Ittai Abraham
Intel Labs, Petah Tikva, Israel

Gilad Asharov
Bar-Ilan University, Ramat-Gan, Israel

Anirudh Chandramouli
Bar-Ilan University, Ramat-Gan, Israel

Abstract

The COOL protocol of Chen (DISC'21) is a major advance that enables perfect security for various tasks (in particular, Byzantine Agreement in Synchrony and Reliable Broadcast in Asynchrony). For an input of size L bits, its communication complexity is $O(nL + n^2 \log n)$, which is optimal up to a $\log n$ factor. Unfortunately, Chen's analysis is rather intricate and complex.

Our main contribution is a simple analysis of a new variant of COOL based on elementary counting arguments. Our main consistency proof takes less than two pages (instead of over 20 pages), making the COOL protocol much more accessible. In addition, the simple analysis allows us to improve the protocol by reducing one round of communication and reducing the communication complexity by 40%.

In addition, we suggest a new way of extracting the core properties of COOL as a new primitive, which we call Graded Dispersal. We show how Graded Dispersal can then be used to obtain efficient solutions for Byzantine Agreement, Verifiable Information Dispersal, Gradecast, and Reliable Broadcast (in both Synchrony and Asynchrony, where appropriate). Our improvement of COOL directly applies here, and we improve the state-of-the-art in all those primitives by reducing at least one round and 40% communication.

2012 ACM Subject Classification Security and privacy; Security and privacy → Cryptography; Security and privacy → Information-theoretic techniques; Security and privacy → Distributed systems security

Keywords and phrases Byzantine Agreement, Broadcast

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.1

Funding *Gilad Asharov*: Supported by the Israel Science Foundation (grant No. 2439/20), and by JPM Faculty Research Award.
Anirudh Chandramouli: Same as Gilad Asharov.

1 Introduction

Byzantine agreement (BA), introduced by [27, 22], is a fundamental problem in distributed protocol design. It involves n parties, each holding an initial input v_i, who must agree on a common value v. BA requires satisfying two key properties: **Agreement**, where all honest parties must output the same value, and **Validity**, which ensures that if all honest parties begin with the same initial value v, their output should also be v. These security properties must be guaranteed even in the presence of up to t corrupted parties, which may deviate arbitrarily from the protocol's specifications. This work focuses on perfect security with optimal resilience ($t < n/3$).

It is known that agreement on a single bit requires $\Omega(n^2)$ bits of communication when the protocol is deterministic [16], or even randomized, against strongly-adaptive adversaries [3] (for any $t = O(n)$). Moreover, agreeing on an L bit message requires $\Omega(nL)$ communication,


© Ittai Abraham, Gilad Asharov, and Anirudh Chandramouli;
licensed under Creative Commons License CC-BY 4.0

since we might have $O(n)$ parties that do not hold the agreed message as input and to receive that message. Furthermore, it has been shown that for any Byzantine agreement protocol with perfect security, there exists an execution that requires $t+1$ rounds [18]. These impossibilities imply that the best one can hope for is $O(nL + n^2)$ for agreeing on a message of size L. However, achieving this bound has been an elusive goal for extensive research.

The baseline protocol [10] for binary Byzantine agreement has a communication complexity of $O(n^2)$ and runs in $\Theta(n)$ rounds. For messages of size L, this implies a communication complexity of $O(n^2L)$. Liang and Vaidya [23] proposed a protocol with a communication complexity of $O(nL + n^4\sqrt{L} + n^6)$ communication; Ganesh and Patra [19], and Loveless, Dreslinski, and Kasikci [24] introduced protocols with a complexity of $O(nL + n^4 \log n)$ communication; Subsequently, Nayak, Ren, Shi, Vaidya, and Xiang [25] improved on this with a protocol that achieves $O(nL + n^3 \log n)$ communication.

The COOL protocol. The paper of Chen [13] further advanced the state of the art by presenting a *deterministic* byzantine agreement protocol, named COOL (stands for **co**ded BA protoc**ol**), that achieves $O(nL + n^2 \log n)$ communication complexity, which is only a logarithmic factor away from the best possible one can hope for. He shows:

▶ **Theorem 1.1** (Byzantine agreement [13]). *There exists a perfectly secure and deterministic Byzantine agreement protocol over a synchronous network for messages of size L bits with communication complexity of $O(nL+n^2 \log n)$ with $\Theta(n)$ rounds, resilient to $t < n/3$ malicious faults.*

Unfortunately, the analysis in [13] is complex and lengthy, making it difficult to understand. Our goal is to provide a simplified analysis of the protocol, offering new insights into the technique and its limitations. The COOL approach has already been applied to related primitives, such as broadcast and gradecast [30] in the synchronous settings, and reliable broadcast in asynchronous settings [6], and holds promise for broader applications as well.

1.1 Our Contributions

In this paper, we provide the following contributions:

Graded dispersal. The COOL protocol of [13] is tailored for Byzantine agreement, and its core was later adopted for reliable broadcast [6] and gradecast by [30]. We show that this core of COOL is essentially a new variant of dispersal (a part of a Verifiable Information Dispersal protocol [12]). We call this new variant *graded dispersal*. We note that dispersal typically needs a designated sender; in our abstraction, we assume the parties have already received the message from some "virtual sender" and wish to verify that they have enough information to retrieve and agree on it later. Defining this graded dispersal variant allows us to distill the novelty of COOL's protocol consistency check and build applications that are based on COOL in a more modular fashion.

Simple analysis. The analysis in [13] is rather involved and relies on various tools; citing [13] "the proof borrows tools from coding theory, together with graph theory and linear algebra". As mentioned above, this also makes the result hard to verify. We provide a new analysis of our graded dispersal protocol. Notably, our analysis is significantly more concise, much shorter than the original COOL analysis, and uses only simple counting. Given the centrality of the result and its simple analysis, we believe that this protocol will be a good candidate for being part of a curriculum in distributed computing classes as a prime example of the centrality of polynomials in obtaining efficient perfect security.

Applications. We show how to apply graded dispersal to achieve the following (1) gradecast (synchrony); (2) multi-valued Byzantine agreement; (3) broadcast; and (4) reliable broadcast (asynchrony).

Gradecast introduced by Feldman and Micali [17] ("graded-broadcast") is a fundamental primitive used for various consensus algorithms.

Bracha's Reliable Broadcast [11] is a useful building block in asynchronous protocols. The recent paper of Das, Xiang, and Ren [15] has a detailed survey of the usefulness of communication-efficient Reliable Broadcast. In particular, it implies Asynchronous Verifiable Secret Sharing and Asynchronous Distributed Key Generation [4].

One round less. Our new protocol reaches agreement faster than the COOL protocol. Specifically, while COOL requires three rounds of reporting agreements to one another (see more in-depth in Section 2), we show that two rounds suffice. This improvement becomes particularly significant when utilizing the COOL technique in constant round protocols such as reliable broadcast and gradecast - and improves the state-of-the-art for those primitives.

Less communication. Our new protocol improves the communication complexity by 40%. Specifically, the protocol of COOL divides the messages into blocks of polynomials of degree $d = t/5$. Our protocol works when dividing the messages into blocks of degree $d = t/3$. Changing the degree enhances the state-of-the-art in all our applications – multi-valued byzantine agreement, broadcast, gradecast, and reliable broadcast.

We consolidate the costs of gradecast and reliable broadcast in Table 1.

Table 1 The communication complexity (in bits) and round complexity of our gradecast and reliable broadcast compared to previous works. We compare here protocols with perfect security and optimal resilience only (see Section 2.3 for other settings). The reported numbers of rounds are for non-balanced protocols; Making the protocols balanced increases the number of rounds by one (except for [17, 11], which remain the same).

Gradecast (synchrony)

	Communication	Rounds
[17]	$O(n^2L)$	2
[2]	$O(nL + n^3 \log n)$	9
[30]	$O(nL + n^2 \log n)$	8
This work	$O(nL + n^2 \log n)$	5

Reliable Broadcast (asynchrony)

	Communication	Rounds
[11]	$O(n^2L)$	4
[26]	$O(nL + n^4 \log n)$	11
[25]	$O(nL + n^3 \log n)$	7
[6]	$O(nL + n^2 \log n)$	9
This work	$O(nL + n^2 \log n)$	6

2 Technical Overview

In this section, we provide a technical overview of the different primitives. We denote by n the number of parties and by t the number of corrupted parties. We assume a finite field $\mathbb{F}$ of size at least $n + 1$ (and for simplicity, $\mathsf{poly}(n)$). We let $t < n/3$ bound the number of corrupted parties (corresponding to optimal resilience in the plain model, that is, in the absence of a PKI).

We consider both synchronous communication, in which protocols proceed in rounds (Section 2.1), and asynchronous communication networks (Section 2.2). Messages between honest parties in the asynchronous network are guaranteed to be delivered, but the adversary can introduce arbitrary delays.

2.1 Synchrony

The COOL protocol in a nutshell. The protocol of COOL divides the value v_i of size L into blocks, where each block is a polynomial of degree-d over a finite field $\mathbb{F}$. For simplicity of exposition, assume that $L = (d+1)\log|\mathbb{F}|$ (i.e., we have a single block – one polynomial).

Two parties P_i and P_j that wish to check whether their polynomials $f_i(x), f_j(x)$ are identical can do so efficiently using randomization and with a statistical error based on the Schwartz-Zippel lemma [29, 31]. Each pair evaluates its respective polynomials on a random point $r \in \mathbb{F}$ and verify that $f_i(r) = f_j(r)$. This approach was taken in [1] for gradecast and reliable broadcast. The novelty of COOL protocol is that it achieves a similar effect of equality check of polynomials, *deterministically*, and *error free*. Each pair of parties exchange just $O(1)$ points, as in the statistical case. Specifically, P_i sends to P_j the points $f_i(i)$ and $f_i(j)$ only, and P_j verifies that those two points agree with its polynomial $f_j(x)$. If a party agrees with $n-t$ parties - then the party is happy. The protocol proceeds in a few rounds, during which each party reports to others whether it is happy. A party remains happy if at least $n-t$ parties in its agreement set report they are happy. Interestingly, the proof of COOL shows that after three rounds of repeating this process, all parties that report that they are happy must hold the same polynomial.

Graded dispersal. We formalize a variant of dispersal, which, as mentioned, distills the consistency property achieved by COOL. We require the following properties from a graded dispersal protocol. Each party holds some polynomial $f_i(x)$ as input (of degree-d). The parties output $(f'_i(x), g_i)$ where $f'_i(x)$ is a polynomial of degree-d (might be $\perp$), and g_i is a grade in $\{0,1,2\}$. The guarantees are: (1) Validity: If all honest parties start with the same polynomial $f(x)$, then they all must output $(f(x), 2)$; (2) Graded agreement: If some honest party outputs $(f(x), 2)$ then at least $t+1$ honest party output $f(x)$ with a grade at least 1, and all other honest parties output $(\perp, 0)$. Note that we have no guarantee in the case where no honest party outputs grade 2, i.e., in that case, different honest parties might have different polynomials as output (with grade 1). Furthermore, our formalization of graded dispersal allows two honest parties to output grades of 2 and 0, respectively. This is different from the typical definition of graded primitives (such as gradecast), where if an honest party outputs grade 2, then all honest parties output a grade at least 1. However, looking ahead, we show that this weak formalization of graded dispersal suffices for all the applications we consider, including reaching a full byzantine agreement.

We implement graded-dispersal based on the COOL protocol. The crux of our contribution is in providing a simple proof for this construction. Moreover, we show we can achieve it in one round of reporting happiness less than COOL. Specifically, we show that:

1. Initially, only two sets of parties, A and B, holding polynomials $f_A(x)$ and $f_B(x)$, respectively, can report that they are happy after the initial pairwise checks. Since two polynomials might agree on only d common points, three parties holding distinct polynomials cannot have enough agreements and cannot all be happy. See Claim 3.4.
2. We show that if there is one set of parties A, all holding the same polynomial $f_A(x)$ and $|A| \geq t+1$, then only parties from A can output grade at least 1, and all other parties output $(\perp, 0)$. Intuitively, if another set of parties B with $|B| \leq t$ holds polynomial $f_B(x)$, then those parties cannot remain happy due to inconsistencies with the larger set A. See Claim 3.5. Importantly, a party outputs grade 2 if it is happy and it received at least $2t+1$ reports from its agreement set in the last round. This implies that it received at least $t+1$ reports from honest parties. In that case, a party can output grade 2, and the other happy parties must also hold $f_A(x)$, and we have at least $t+1$ parties with the same (and only) output polynomial, $f_A(x)$.

3. If no large set exists, no party outputs grade 2 (no party is happy and received $2t+1$ reports of happiness from its agreed set in the last round). See Claim 3.6.

Our graded dispersal works in three rounds and requires $O(nL+n^2\log n)$ for inputs of size L.

Data dissemination. In Section 3.2, we show a protocol that allows parties to reconstruct the output $f(x)$ once there are at least $t+1$ honest parties that hold the same input $f(x)$ (and other honest parties hold $\perp$). Essentially, this is precisely the guarantee we have once one party outputs 2 in the graded dispersal protocol. Simply, each party that has an input $f(x)$ sends to each party P_i its point $f(i)$. P_i then takes as its point the majority value that it received, and since $t+1$ honest parties that hold $f(x)$ and there are at most t corrupted parties, we are guaranteed that the majority of P_i is $f(i)$. Once each party P_i has a point $f(i)$, the parties reconstruct f by sending their points to one another and using Reed-Solomon decoding.

Given graded dispersal and data dissemination, we provide three applications in synchrony.

Application I: Gradecast. Gradecast is a relaxation of broadcast and involves a sender that holds a message M, and all parties output (M_i, g_i) with $g_i \in \{0,1,2\}$. If the sender is honest, all parties must output $(M,2)$. The guarantee is that if one honest party outputs $(M,2)$, all honest parties must output (M,g_i) with $g_i \geq 1$. We remark that the definition that we provide is a relaxation of Gradecast that was formalized by Katz and Koo [21], which allows the parties to output different messages when there is no honest party with grade 2 (just as in our graded dispersal). We are unaware of any application of gradecast in which the relaxed definition does not suffice.

We show how to implement gradecast using graded dispersal and data dissemination. Specifically, the sender sends its message to all parties, and the parties then run graded-dispersal. After that, they run data dissemination and output the message they received in the data dissemination with their grade in the graded dispersal. It is easy to see that the requirements for gradecast are met.

The last round of graded dispersal can be run parallel to the first round of data dissemination, so we can get a protocol that runs in a total of 5 rounds. This results in an improvement of 3 rounds compared to the state-of-the-art [30]; yet, our guarantee is slightly weaker. We also show how to satisfy the [17] definition at the expense of adding one more round. We then get a strict improvement of 2 rounds with the same security definition (and reduce communication by 40%). The improvement of 2 rounds is due to some redundancy in the protocol of [30].

We emphasize that the "relaxed" gradecast suffice for all the applications we are aware of, specifically to achieve broadcast with an expected constant number of rounds (see Section 2.3). We therefore advocate using the relaxed version.

Application II: Multi-Valued Byzantine Agreement. We compose graded-dispersal, with binary agreement (on whether a party has grade 2), and data dissemination to achieve multi-valued byzantine agreement. As mentioned, our protocol has one less round than the state-of-the-art and 40% less communication. The protocol runs in $O(nL+n^2\log n)$ with $\Theta(n)$ rounds (due to the binary agreement).

Application III: Broadcast. Again, by composing graded-dispersal, binary agreement, and data dissemination, we obtain a broadcast protocol that runs in $O(nL+n^2\log n)$ communication and $\Theta(n)$ rounds.

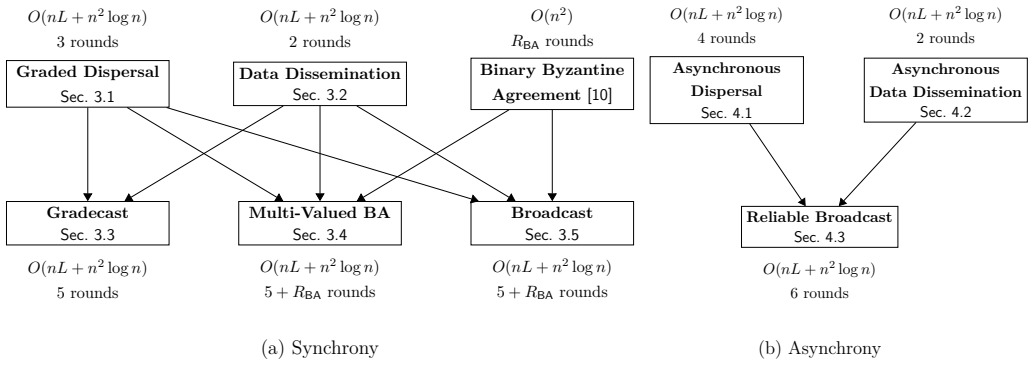


Figure 1 A roadmap of the paper, and the different primitives we construct. Next to each primitive, we report the round complexity and the communication complexity. When reporting the number of rounds – we report for a non-balanced protocol variation. Making each protocol balanced requires one more round of communication.

For gradecast and broadcast, we also show a variant where the protocol is balanced: each party sends or receives $O(L + n \log n)$, and there is no party whose communication load is higher than the others. This comes at the expense of adding an extra round.

2.2 Asynchrony

In reliable broadcast, there is a sender who holds a message M. If the sender is honest, then the protocol always terminates, and all honest parties must output M. If the sender is corrupted and the honest parties terminate, then they must all output the same message M'. Note that the protocol might not terminate if the sender is not honest.

Based upon graded dispersal in synchrony, we show a dispersal protocol in asynchrony (Section 4.1). We show perfect asynchronous data dissemination which has a weak agreement property. Our protocol and formalization is identical to that of [14] and we present it here for completeness of our result.

We have to add one more round than synchrony to guarantee the termination of honest parties for the honest sender case. We also provide an asynchronous variant of data dissemination (Section 4.2). Combining those primitives, we obtain a reliable broadcast.

Our reliable broadcast is deterministic and runs in 6 rounds. This is an improvement of 3 rounds, in addition to 40% reduced communication, compared to [6] that is based on COOL. See Section 4.3 for the protocol, and Table 1 for the improvements over previous protocols.

2.3 Related Work

All our protocols are deterministic and error-free. The applications we are considering (broadcast, multi-valued BA, gradecast, and reliable broadcast) were also studied in different settings.

Protocols with an expected number of rounds. As mentioned, broadcast and byzantine agreement in that setting must incur $\Omega(n)$ rounds [18]. The broadcast and byzantine agreement presented in our paper achieves $O(n)$ rounds in the worst case. Rabin [28] and Ben-Or [8] studied the effect of randomization on round complexity, and Feldman and Micali [17] gave the first protocol with an expected constant round protocol for byzantine agreement with optimal resilience. Many works have improved the communication complexity

of expected constant round Byzantine agreement protocols with optimal resilience [20, 25, 2, 7] where the current state of the art is $O(nL + n^3 \log^2 n)$ with expected constant number of rounds. We remark that in all those protocols, the sender first gradecasts its message, and therefore using our improved gradecast (whose relaxed notion suffices for this application) improves the constant in the "expected constant" of these works.

Gradecast. Gradecast is an important primitive used for various consensus algorithms, e.g., multi-consensus, approximate agreement [9], or Phase-King [5]. Our round complexity is the most efficient in the perfect setting. However, the protocol of Abraham and Asharov [1] is one round better at the expense of introducing some error (i.e., statistical security).

Reliable broadcast. In the computational setting, the protocol of Cachin and Tessaro [12] achieves $O(nL+\kappa n^2 \log n)$ where κ is the computational security parameter. This is improved by Alhaddad et al.[6] to $O(nL + n^2\kappa)$ with 4 rounds of communication. The recent work by Locher and Shoup looks into the constants and obtains $1.5nL + O(\kappa n^2 \log n)$.

Organization

The rest of the paper is organized as follows. The synchronous protocols appear in Section 3, and the asynchronous protocols appear in Section 4. See also Figure 1 for the previous primitives and their interplay.

3 Synchrony

In this section, we provide the protocols for synchronous communication.

Preliminaries. In the analysis, we sometimes use $H \subseteq [n]$ to denote the set of honest parties and $I \subseteq [n]$ to denote the set of corrupt parties. Our protocols also use Reed Solomon decoding. Let $R = \{(i, v_i)\}$ denote a set of evaluations of a degree d polynomial over $\mathbb{F}$ where up to t of the points may be incorrect. If $|R| > 2t + d$, then there exists an efficient algorithm that upon input R, d and t, can recover a degree-d polynomial $f(x)$ such that for $|R| - t$ points $(i, v_i) \in R$, it holds that $f(i) = v_i$. We denote this algorithm as $\mathsf{RSDec}(R)$.

Communication complexity. In each of our protocols, we assume that the input (either of the designated sender or each one of the parties) is a polynomial of degree at most d over the field $\mathbb{F}$. For the general case of an L-bit input, the parties segment the L-bit message into $\ell = \left\lceil \frac{L}{(d+1)\log|\mathbb{F}|} \right\rceil$ blocks of $(d+1)\log|\mathbb{F}|$ bits each (with appropriate padding for the last block if L is not a multiple of $(d+1)\log|\mathbb{F}|$). The parties then execute ℓ instances of the same protocol, where the input for the mth instance is the mth block interpreted as a polynomial of degree at most d. Aggregation over the ℓ different executions is performed in a natural way, e.g., a party P_i agrees with P_j if and only if they agree in all the ℓ instances. We, therefore, omit such details. We report the communication complexity of our protocols over inputs of size L bits while we present the protocols assuming $|L| = (d+1)\log|\mathbb{F}|$. Additionally, note that $\log|\mathbb{F}| \in O(\log n)$ and our communication complexity is reported in that light.

The degree d. We set the degree of our polynomials to be $d = \lfloor t/3 \rfloor$. Looking ahead, we remark that the only place this is necessary (except for requiring $d \leq t$) is in the proof of Claim 3.4.

3.1 Building Block I: Graded Dispersal

We introduce the notion of graded dispersal:

▶ **Definition 3.1** (Graded Dispersal). *A protocol for parties $P_1, \ldots, P_n$ where*

- **Input:** *Each party P_i holds as input some polynomial $f_i(x)$ of degree d over $\mathbb{F}$;.*
- **Output:** *An output $(f_i(x), g_i)$ with grade 2 or 1, or an output $(\perp, 0)$.*

is graded dispersal protocol tolerating t malicious parties *if the following properties hold:*

- **Validity:** *If all honest parties start with the same polynomial $f(x)$, then all output $(f(x), 2)$.*
- **Weak Graded Agreement:** *If an honest party outputs $(f(x), 2)$, then at least $t+1$ honest parties output $(f(x), g_i)$ with $g_i \geq 1$, and all other honest parties output $(\perp, 0)$.*

Note that if there is no honest party with grade 2 then honest parties with non-$\perp$ output and grade 1, may disagree on their output value. Moreover, grade 2 does not imply full agreement among the honest parties.

Nevertheless, we show that this Weak Graded Agreement property is sufficient to work well with the Data Dissemination protocol (see Section 3.2).

Protocol 3.2: Graded Dispersal

Input: Each party P_i holds $f_i(x)$ of degree at most $d = \lfloor t/3 \rfloor$ over $\mathbb{F}$ to encode $d \log n$ bits.
The protocol:

1. **Exchange:**
   a. P_i sends $(f_i(i), f_i(j))$ to each P_j.
2. **Dynamic set A_i^1:**
   a. Let (u_j, v_j) be the two values that P_i receives from P_j. If $f_i(j) = u_j$ and $v_j = f_i(i)$ then add j to A_i^1.
   b. If $|\mathsf{A}_i^1| \geq n - t$ then send OK_1 to all parties.
3. **Dynamic set A_i^2:**
   a. If OK_1 is received from P_j for $j \in \mathsf{A}_i^1$ then add j to A_i^2.
   b. If $|\mathsf{A}_i^2| \geq n - t$ then send OK_2 to all parties.
4. **Output:**
   a. If sent OK_2, and received at least $2t+1$ messages OK_2, then output $(f_i(x), 2)$.
   b. If sent OK_2, but received $\leq 2t$ messages OK_2, then output $(f_i(x), 1)$.
   c. Otherwise, output $(\perp, 0)$.

In the COOL protocol, there is an additional round of OK_3 messages and the degree is $d = \lfloor t/5 \rfloor$. In our protocol, there are just two rounds of OK and the degree is $d = \lfloor t/3 \rfloor$. Hence our protocol reduces the number of rounds from 3 to 2 (by %33) and reduces the overhead per bit from 15 to 9 (by %40). Note that when $L > d \log n$, only the first round is parallelized.

▶ **Theorem 3.3.** *Protocol 3.2 is a graded-dispersal protocol for messages of size $L = d \log n$ bits where $d = \lfloor t/3 \rfloor$ that is resilient to $t = \lfloor (n-1)/3 \rfloor$ malicious parties. The protocol runs in 3 rounds and incurs a total communication of $\leq \binom{n}{2} \log n + 2\binom{n}{2} < n^2 \log n$ bits.*

Proof. We now show that all the properties hold as in Definition 3.1.

Validity. Assume that all honest parties have the same input $f(x)$. For each party P_i, the set A_i^1 will contain the set of all honest parties and therefore $|\mathsf{A}_i^1| \geq n - t$ and P_i sends OK_1. Moreover, it will receive OK_1 from all honest parties; therefore, P_i also sends OK_2. We get that all honest parties send OK_2, and consequently, all receive at least $2t + 1$ OK_2, so all output $(f(x), 2)$.

Weak graded agreement. If one party outputs $(f(x), 2)$, then there are at least $t+1$ honest parties that output $(f(x), g)$ with $g \geq 1$, and all the others output $(\perp, 0)$. This is done in 3 steps:

- Claim 3.4 proves that there cannot be three honest parties that hold different inputs, that all send OK_1. Hence there exists at most two inputs f_a, f_b such that each honest party that sends OK_1 must hold one of these two inputs.
- Given this, Claim 3.5 proves that if there is a set of at least $t + 1$ honest parties that all have the same input, then no honest party holding a different input will send OK_2.
- Claim 3.6 shows that if there is no set with at least $t + 1$ honest parties that have the same input, then no honest party outputs grade 2.

Proof of Weak graded agreement given the Claims 3.4, 3.5, 3.6. If an honest party outputs grade 2, then from Claim 3.6 we must be in the case where there are at least $t + 1$ honest parties as in Claim 3.5 and from that all honest parties that send OK_2 (hence output grade 1) must output the same value.

Proving Claims 3.4, 3.5, 3.6 concludes the proof of the Theorem.

▷ Claim 3.4. There can be at most two distinct inputs, such that there exist honest parties that hold these inputs and send OK_1 messages.

Proof. Seeking a contradiction, assume there exist honest parties P_a, P_b, P_c that hold distinct input polynomials f_a, f_b, and f_c, such that P_a, P_b, P_c all send OK_1 messages.

Let H be the set of all honest parties and let $I = [n] \setminus H$ be the malicious parties.

For any $x \in \{a, b, c\}$ define S_x as the set of all the honest parties that agree with P_x:

$$S_x := \mathsf{A}_x^1 \cap H \ .$$

Observe that:

- For any $x \in \{a, b, c\}, |S_x| \geq n - t - |I|$. Because if P_x sent OK_1, it must agree with at least $n - t$ parties, and thus with at least $n - t - |I|$ honest parties.
- For any $x, y \in \{a, b, c\}, |S_x \cap S_y| \leq d$. Because for any $j \in S_x \cap S_y$ then $f_x(j) = f_j(j) = f_y(j)$. So if $|S_x \cap S_y| \geq d+1$ then f_x and f_y agree on at least $d+1$ points, and therefore, must be identical.

So from inclusion-exclusion:

$$\begin{aligned} |S_a \cup S_b \cup S_c| &= |S_a| + |S_b| + |S_c| - |S_a \cap S_b| - |S_a \cap S_c| - |S_b \cap S_c| + |S_a \cap S_b \cap S_c| \\ &\geq 3(n - t - |I|) - 3d + 0 \ . \end{aligned}$$

Since $d = \lfloor t/3 \rfloor$ so $3d \leq t$:

$$|S_a \cup S_b \cup S_c| \geq 3n - 3t - 3|I| - t = 3n - 4t - 3|I| \ .$$

On the other hand, $(S_a \cup S_b \cup S_c) \subseteq H$, and therefore $|S_a \cup S_b \cup S_c| \leq |H| = n - |I|$:

$$3n - 4t - 3|I| \leq |S_a \cup S_b \cup S_c| \leq n - |I| \ .$$

Using $n \geq 3t+1$ and $|I| \leq t$:

$$\begin{aligned} 2n-4t &\leq 2|I| \\ 2(3t+1)-4t \leq 2n-4t &\leq 2|I| \leq 2t \\ 2t+2 &\leq 2t\ , \end{aligned}$$

which is a contradiction. ◁

So let T_a and T_b be the at most two sets of honest parties with inputs f_a and f_b respectively such that only members in those sets send OK_1. Since $\mathsf{A}_j^2 \subseteq \mathsf{A}_j^1$ for each honest party P_j then only parties in T_a or T_b may send OK_2.

Assume wlog that $|T_a| \geq |T_b|$.

Define:

$$T_{a+} = \{i \in T_a \mid f_a(i) = f_b(i)\}\ , \qquad T_{a-} = T_a \setminus T_{a+} \quad (= \{i \in T_a \mid f_a(i) \neq f_b(i)\})\ ,$$
$$T_{b+} = \{i \in T_b \mid f_a(i) = f_b(i)\}\ , \qquad T_{b-} = T_b \setminus T_{b+} \quad (= \{i \in T_b \mid f_a(i) \neq f_b(i)\})\ .$$

▷ Claim 3.5. If $|T_a| \geq t+1$ then no party in $H \setminus T_a$ sends OK_2.

Proof. From Claim 3.4, honest parties not in $T_a \cup T_b$ do not send OK_1.

Observe that parties in T_{b-} do not send OK_1. This is because $|T_a| \geq t+1$, all $j \in T_{b-}$ have $f_a(j) \neq f_b(j)$ by definition. Therefore, any $j \in T_{b-}$ cannot receive support from T_a and thus

$$|\mathsf{A}_j^1| \leq n - |T_a| \leq n - t - 1 < n - t\ .$$

Given this, parties in in T_{b+} do not send OK_2. This is because the only honest parties that can send OK_1 that will be accepted by T_{b+} are from $T_{a+} \cup T_{b+}$, but $|T_{a+} \cup T_{b+}| \leq d$ because otherwise $f_a = f_b$. So for any party $j \in T_{b+}$ must have

$$|\mathsf{A}_j^2| \leq |T_{a+}| + |T_{b+}| + |I| \leq d + t \leq 2t < n - t\ .$$ ◁

▷ Claim 3.6. If $|T_a| \leq t$ then no party outputs grade 2.

Proof. Observe that the parties in T_{a-} and T_{b-} do not send OK_2. This is because all parties not in T_a, T_b do not send OK_1, and for $x \in \{a, b\}$, a party in T_{x^-} can receive OK_1 only from T_x and the corrupted parties. Thus,

$$|\mathsf{A}_x^2| \leq |T_x| + |I| \leq t + t = 2t < n - t\ .$$

Given this, only parties in T_{a+}, T_{b+} might send OK_2. However, no party can receive $2t+1$ OK_2 messages. This is because $|T_{a+} \cup T_{b+}| \leq d$ so

$$|\mathsf{A}_x^2| \leq |T_{a+}| + |T_{b+}| + |I| \leq d + t \leq 2t < n - t\ .$$ ◁

This concludes the proof of Theorem 3.3. ◀

Note: adding one more round. We are unaware of an application that actually requires this a property, where grade 2 implies that all parties output the same value with grade > 2 (so graded agreement not just weak graded agreement as in Definition 3.1). Nevertheless, we comment here that can be obtained by one more round.

To achieve this, parties that receive $n - t$ OK_2s send OK_3, and then: (1) if P_i sent OK_3 and received $2t+1$ OK_3 it outputs $(f(x), 2)$; (2) otherwise, if it sent OK_3 it outputs $(f(x), 1)$; (3) otherwise, (it did not send OK_3) it outputs $(\perp, 0)$.

If all parties start with the same input $f(x)$ then they all send and receive $n-t$ OK_3s and all output $(f(x), 2)$. In addition, if there is no set $|T_a| \geq t+1$, then all honest parties must output $(\perp, 0)$. In this case, Claim 3.6 shows that no party receives $2t+1$ OK_2; therefore, no honest party sends OK_3.

3.2 Building Block II: Data Dissemination

▶ **Definition 3.7.** *A protocol for parties $P_1, \ldots, P_n$ is where each party P_i has an input $f_i(x)$ (of degree d over field $\mathbb{F}$, might be $\perp$) and an output $f'_i(x)$, is* data dissemination protocol tolerating t malicious parties *if it satisfies the following property:*

- **Output consistency:** *If there exists a polynomial $f^*(x)$ such that there are at least $t+1$ honest parties with the input $f^*(x)$, and all other honest parties start with input $\perp$, then all honest parties output $f^*(x)$.*

Protocol 3.8: Data Dissemination

Input: Each party P_i holds $f_i(x)$ as input, of degree at most d. Some P_i might have input $\perp$.
The protocol:

1. **Round I – Exchange:** If $f_i(x) \neq \perp$, send to each P_j its point $f_i(j)$.
2. **Round II – Set and send your point:** Each party P_i receives messages $(u_1, \ldots, u_n)$. Let u_i be the value received at least $t+1$ times. Send u_i to all parties.

Output: Let $(y_1, \ldots, y_n)$ be the received values. Run Reed Solomon decoding RSDec on $(y_1, \ldots, y_n)$ to find a unique polynomial of degree-d $f'_i(x)$ with at most t errors. Output $f'_i(x)$. If there is no unique decoding, output $\perp$.

▶ **Theorem 3.9.** *Protocol 3.8 is a data-dissemination protocol tolerating $t < n/3$ malicious parties (as per Definition 3.7). The protocol takes two rounds of communication and $O(nL + n^2 \log n)$ communication, where each party starts with an input of size L.*

Proof. Assume that $t+1$ honest parties start with the same input $f(x)$. Each honest party P_j receives $f(j)$ at least $t+1$ times. All other honest parties that hold $\perp$ send nothing in the first message. As such, the majority value of P_j must be $f(j)$, and it sends it to all other parties in the second round. Therefore, all honest parties must receive $(y_1, \ldots, y_n)$ that is of distance at most t from $(f(1), \ldots, f(n))$. Since $n > 2t + d$, the Reed Solomon decoding guarantees that all output $f(x)$.

Protocol 3.8 as described above incurs a total communication of $O(n^2 \log n)$ bits and this corresponds to the case when each party starts with an input of size $(d+1)\log|\mathbb{F}|$ bits. Hence, the communication complexity of Protocol 3.8, where each party starts with inputs of size L bits, is $O(nL + n^2 \log n)$ bits. ◀

3.3 Application I: Gradecast

We now show how the above protocols can be utilized to achieve gradecast.

Gradecast is a relaxation of broadcast introduced by Feldman and Micali in 1988 [17]. Our protocol follows a relaxed notion of Gradecast formalized by Katz and Koo [21].

▶ **Definition 3.10.** *A protocol for parties $P_1, \ldots, P_n$, where a distinguished sender $P^* \in \{P_1, \ldots, P_n\}$ holds an initial input M is* a gradecast protocol tolerating t malicious parties *if the following conditions hold for any adversary controlling at most t parties:*

- *Each honest party P_i outputs a message m_i and a grade $g_i \in \{0, 1, 2\}$.*
- *If the sender is honest, then the output of every honest party P_i satisfies $m_i = M$ and $g_i = 2$.*
- *If there exists an honest party P_i which outputs a message m_i and the grade $g_i = 2$, then the output of every honest party P_j satisfies $m_j = m_i$ and $g_j \geq 1$.*

Protocol 3.11: Gradecast from Graded-Dispersal and Data Dissemination

Input: The sender P^* holds $f(x)$ of degree at most d. Other parties have no input.
The protocol:

1. **The sender:** The sender sends $f(x)$ to all parties.
2. **Each party P_i:** The parties invoke the graded-dispersal protocol (Protocol 3.2) where P_i inputs $f_i^{(1)}(x)$. Let $(f_i^{(2)}(x), g_i)$ be the output of the graded-dispersal protocol.
3. **Each party P_i:** The parties run the data-dissemination protocol (Protocol 3.8) with input $f_i^{(2)}(x)$. Let $f_i^{(3)}(x)$ be the output.
4. **Output:** If $g_i = 2$, then output $(f_i^{(3)}(x), 2)$. Otherwise, if $f_i^{(3)}(x) \neq \perp$, then output $(f_i^{(3)}(x), 1)$. Otherwise, output $(\perp, 0)$.

▶ **Theorem 3.12.** *Protocol 3.11 is a Gradecast protocol tolerating up to $t < n/3$ malicious parties (as per Definition 3.10). The protocol takes 5 rounds and $O(nL + n^2 \log n)$ communication for gradecasting a message of size L.*

Proof. We prove the case of an honest sender and a corrupted sender.

An honest sender. We first show that if the sender is honest and has an input $f(x)$, then all honest parties output $(f(x), 2)$. Specifically, all honest parties start the graded dispersal with the same input $f(x)$. Therefore, from the validity of graded-dispersal, all honest parties output $(f(x), 2)$. Moreover, since more than $t + 1$ honest parties hold the same input $f(x)$ in the data-dissemination protocol, all honest parties output $f(x)$ in that protocol, and output $(f(x), 2)$ in the gradecast protocol.

An honest party with output grade 2. We next consider the case where there exists an honest party P_j with an output grade 2. In that case, P_j must have grade 2 in the graded-dispersal protocol. As such, from the graded agreement, there are at least $t + 1$ honest parties that output the same polynomial $f(x)$ with grade $g_i \geq 1$ from graded dispersal. Moreover, from the properties of graded-dispersal, parties with output that is not $f(x)$ use $\perp$ as their input in the data dissemination protocol. Thus, there are at least $t + 1$ honest parties with input $f(x)$ in the data dissemination protocol, and therefore all honest parties must output $f(x)$ in the data dissemination. All honest parties output $f(x)$ with a grade of at least 1. ◀

Reducing one-round. We can reduce one round of interaction by sending the first round of the data-dissemination protocol together with the last round of the graded dispersal protocol. Specifically, each party that sends OK_2 message to P_j (the last round of the weak dispersal) sends together with it the point $f_i(j)$ to party P_j (the first message of the data dissemination).

We, therefore, get that the total round complexity of the protocol is five rounds (Theorem 3.12 is reported in that light).

Making the protocol balanced. In the protocol as described, the communication complexity of the sender is $O(n(d+1)\log n)$ while the communication complexity of every other party is just $O(n \log n)$. To reduce the communication complexity for the sender, the sender can first send to each party P_i its point $f(i)$. Each party forwards the point it received to all others, and the parties use Reed Solomon decoding to reconstruct the polynomial $f(x)$.

Stronger property. As mentioned, we use the relaxed definition of gradecast as defined in [20]. The definition by [17] requires that in the absence of a party with grade 2, all parties with grade 1 must agree on their message as well. This can be achieved at the expense of adding one round to the graded-dispersal, as discussed in Section 3.1.

3.4 Application II: Multi-Valued Byzantine Agreement

▶ **Definition 3.13** ((Binary/Multi-valued) Byzantine Agreement). *A protocol for parties $P_1, \ldots, P_n$ where each party P_i holds initial input v_i, is a* Byzantine agreement protocol tolerating t malicious parties *if the following conditions hold for any adversary controlling at most t parties:*

- **Agreement:** *All honest parties must output the same value.*
- **Validity:** *If all honest parties begin with the same input value v, then all honest parties output v.*
- **Strong consistency:** *If not all honest parties begin with the same input v, then all honest parties output the same value, which was an input of some honest party.*

For brevity, if the values $\{v_i\}$ are restricted to binary values, then we call the protocol a binary Byzantine agreement.

Protocol 3.14: Multi-Valued Byzantine Agreement from Graded Dispersal, Binary Byzantine Agreement, and Data Dissemination

- **Input:** Each party P_i holds a polynomial $f_i^{(1)}(x)$ of degree at most d over $\mathbb{F}$.
- **The protocol:**
  1. The parties invoke the graded dispersal protocol (Protocol 3.2), where each party P_i inputs $f_i^{(1)}(x)$. Let $(f_i^{(2)}(x), b_i)$ be the output.
  2. The parties run the binary Byzantine agreement protocol (e.g., [10]), where the input of each P_i is the bit b_i. Let b be the output of the Byzantine agreement.
  3. If $b = 1$, then the parties invoke the data dissemination protocol (Protocol 3.8) where each party P_i inputs $f_i^{(2)}(x)$. Let $f_i^{(3)}(x)$ be the output of P_i.
     If $b = 0$, then P_i sets $f_i^{(3)}(x) = \perp$.
- **Output:** Each P_i outputs $f_i^{(3)}(x)$.

We recall that the binary Byzantine agreement of [10] requires $O(n^2)$ communication and $\Theta(n)$ rounds for agreeing on a single bit. We get:

▶ **Theorem 3.15.** *When instantiated with a binary Byzantine agreement protocol with $O(n^2)$ communication and $\Theta(n)$ rounds ([10]), protocol 3.14 is a Byzantine Agreement protocol tolerating $t < n/3$ malicious parties (as per Definition 3.13). The protocol uses $O(nL + n^2 \log n)$ communication and $\Theta(n)$ rounds where each party starts with an input of size L.*

Proof. We first show validity, then agreement.

Validity. If all honest parties have the same input $f(x)$, then from the graded agreement property of graded dispersal (Definition 3.1), we get that all honest parties output $(f(x), 2)$. All honest parties then input 1 to the Binary Byzantine agreement, and then all honest parties receive 1 as output. The parties then invoke the data-dissemination protocol, and since there are more than $t+1$ honest parties with the input $f(x)$, all honest parties receive $f(x)$ as output and output it.

Agreement. The output of the parties is determined by the output of the binary byzantine agreement, b, which is the same for all parties. There are two cases to consider:

- If $b = 0$, then all honest parties output $\perp$, and agreement holds.
- If $b = 1$, then there must exists an honest party P_j with an input $b_j = 1$ to the Binary Byzantine Agreement. By the properties of weak-dispersal (Definition 3.1), this implies that at least $t+1$ honest parties have the same output $f(x)$ from the weak-dispersal protocol, i.e., the input of each honest party to the data-dissemination protocol (Protocol 3.8) is either $\perp$ or $f(x)$, and there are at least $t+1$ honest parties with input $f(x)$. The data-dissemination protocol then guarantees that all parties output $f(x)$. The parties output that value in our Byzantine Agreement protocol, and we again have an agreement.

Strong consistency. The only case that is interesting is when $b = 1$. The graded dispersal guarantees that output was also an input of some honest party. ◀

3.5 Application III: Broadcast

▶ **Definition 3.16** (Broadcast). *A protocol for parties $P_1, \ldots, P_n$ where a distinguished sender P^* holds an initial input M, is a* broadcast protocol tolerating t malicious parties *if the following conditions hold for any adversary controlling at most t parties:*

- **Agreement:** *All honest parties output the same value.*
- **Validity:** *If the sender is honest, then all honest parties output M.*

Protocol 3.17: Broadcast from Graded-Dispersal, Data Dissemination, and Binary Agreement

- **Input:** The sender holds a polynomial $f(x)$ of degree at most d over $\mathbb{F}$.
- **The protocol:**
  1. **The sender:** The sender sends $f(x)$ to all parties.
  2. **Each party P_i:** Let $f_i^{(1)}(x)$ be the received polynomial. The parties invoke graded-dispersal protocol (Protocol 3.2), where each party P_i inputs $f_i^{(1)}(x)$. Let $(f_i^{(2)}(x), g_i)$ be the output.
  3. **Each party P_i:** Run Binary byzantine agreement where each party P_i inputs 1 iff $g_i = 2$. Let b be the output.
  4. **Each party P_i:** If $b = 1$, then the parties run the data-dissemination protocol (Protocol 3.8), where each party P_i inputs $f_i^{(2)}(x)$. Let $f_i^{(3)}(x)$ be the output of the data dissemination protocol.
- **Output:** If $b = 1$, then output $f_i^{(3)}(x)$. If $b = 0$ then output $\perp$.

▶ **Theorem 3.18.** *Protocol 3.17 is a broadcast protocol tolerating $t < n/3$ malicious parties (as per Definition 3.16). The protocol uses $\Theta(n)$ rounds and $O(nL + n^2 \log n)$ communication for broadcasting a message of size L.*

Proof. We show agreement and validity.

Agreement. By the binary byzantine agreement, all parties receive the same bit b. If $b = 0$, then all honest parties output $\perp$ and agreement holds; If $b = 1$, then it must be that some honest party received grade 2 in the graded dispersal. As follows from the guarantees of graded dispersal, this implies that there are at least $t + 1$ honest parties with the same output $f(x)$. Those honest parties input $f(x)$ to the data dissemination, and the output of all honest parties is then $f(x)$. Agreement again holds.

Validity. If the sender is honest and starts with input $f(x)$ of degree-d over $\mathbb{F}$, then from the guarantees of graded-dispersal, all honest parties output $(f(x), 2)$. As such, all honest parties input 1 to the binary byzantine agremeent, and must agree on $b = 1$. Therefore, all run the data-dissemination protocol, and all output $f(x)$. ◀

Making the protocol balanced. In the broadcast protocol as described in Protocol 3.17, the communication complexity of the sender is $n(d + 1) \log n$ bits, whereas the communication complexity of every other party is just $O(n \log n)$ bits. As before, to reduce the communication complexity for the sender, the sender can first send to each party P_i its point $f(i)$. The parties then forward the received point to all others and use Reed Solomon decoding to reconstruct the polynomial $f(x)$.

4 Reliable Broadcast in Asynchrony

We now move to asynchronous communication. Here, when an honest party sends a message it is guaranteed to be delivered, but the adversary can introduce arbitrarily delays.

4.1 Dispersal

▶ **Definition 4.1** (Dispersal). *A protocol for parties $P_1, \ldots, P_n$ where the input of each party P_i is some $v_i \in \mathbb{F}$, is an* asynchronous dispersal protocol tolerating t corrupted parties *if the following properties hold:*

- **Termination:** *If one honest party terminates, then all honest parties terminate.*
- **Weak agreement:** *If one honest party terminates with output $f(x)$, then at least $t + 1$ other honest parties output also $f(x)$. Furthermore, all honest parties that terminate with a non-$\perp$ output, terminate with the same polynomial $f(x)$.*
- **Weak Validity:** *If all honest parties start with the same polynomial $f(x)$ of degree at most d, then termination and weak agreement hold with respect to $f(x)$.*

Our asynchronous dispersal protocol is fairly identical to the synchronous counterpart. However, when executed under asynchrony, a key distinction is in the construction of the dynamic sets, $\mathsf{A}_i^1, \mathsf{A}_i^2$. As the adversary may introduce arbitrary delays, the OK_2 messages may not be delivered after the OK_1 messages. The parties construct the A_i^1 and A_i^2 sets and update them in an online fashion upon receiving any message (OK_1 or OK_2).

Protocol 4.2: Dispersal

Input: Each party P_i holds $f_i(x)$ of degree at most d over $\mathbb{F}$.
The protocol:

1. Initialization: Each party initializes $S_i = \emptyset, \mathsf{A}_i^1 = \mathsf{A}_i^2 = \emptyset$, and $f_i(x) = \perp$.
2. **Exchange:**
   a. P_i sends $(\texttt{Exchange}, f_i(i), f_i(j))$ to each P_j.
3. **Dynamic set A_i^1:**
   a. **Upon** receiving $(\texttt{Exchange}, u_j, v_j)$ from P_j, if $f_i(j) = u_j$ and $f_i(i) = v_j$ then add j to A_i^1 .
   b. **Upon** $|\mathsf{A}_i^1| \geq n - t$, send OK_1 to all.
4. **Dynamic set A_i^2:**
   a. **Upon** receiving message OK_1 from P_j for which $j \in \mathsf{A}_i^1$, then add j to A_i^2.
   b. **Upon** $|\mathsf{A}_i^2| \geq n - t$, send OK_2 to all.
5. **Sending** `Done`**:**
   a. If P_i sent OK_2, and **upon** receiving $2t + 1$ OK_2 messages, send `Done` message to everyone.
   b. **Upon** receiving $t + 1$ `Done` messages from distinct parties, send `Done` to everyone.
   c. **Upon** receiving $2t + 1$ `Done` messages, if P_i sent OK_2 message, then terminate and output $f_i(x)$. If P_i did not send OK_2 message, then terminate and output $\perp$.

▶ **Theorem 4.3.** *Protocol 4.2 is an asynchronous dispersal protocol tolerating $t < n/3$ malicious parties (as per Definition 4.1). The protocol takes 4 rounds and a total communication of $O(nL + n^2 \log n)$ bits where each party starts with an input of size L.*

Proof. We show each one of the properties separately.

Termination. An honest party terminates only after it receives $2t + 1$ `Done` messages. This implies that it received at least $t + 1$ messages `Done` from honest parties (and in particular, it also sent the `Done` message). Eventually, all honest parties will receive those $t + 1$ `Done` messages and, in response, will also send a `Done` message. We obtain that eventually, all honest parties send `Done` message, and therefore each honest party will eventually receive $2t + 1$ `Done` messages, regardless of the messages of the adversary. Therefore, all honest parties must terminate.

Weak agreement. An honest party that terminates with an output $f^*(x) \neq \perp$ must have sent an OK_2 message. Moreover, since it terminated, it must have received at least $2t + 1$ `Done` messages, i.e., it received `Done` messages from at least $t + 1$ honest parties. An honest party sends a `Done` message if it either received $2t + 1$ OK_2 messages or received $t + 1$ `Done` messages. The latter implies that at least one honest party sent a `Done` message due to receiving $2t+1$ OK_2 messages, and at least $t+1$ honest parties sent OK_2 message. As follows from Claim 3.4, there are at most two sets of parties (with the same polynomial) that could have sent OK_1. We have two cases to consider:

- As follows from Claim 3.6, if there is no large set of $t+1$ parties with the same polynomial, then no party receives $2t + 1$ OK_2. This implies that no honest party would have sent the `Done` message, and no party terminates.
- If there is a large set of $t+1$ parties with the same polynomial $f^*(x)$, then no other party sends OK_2.

This implies that if some honest party terminates, then we must have a large set of $t+1$ parties with the same polynomial $f^*(x)$. This implies that at least $t+1$ honest parties terminate with that polynomial $f^*(x)$. Any other honest party that terminates with non-$\perp$ must also output $f^*(x)$.

Weak Validity. If all honest parties start with the same polynomial $f^*(x)$, then eventually, all honest parties will appear in the agreed sets of one another. As such, eventually, all honest parties will send the OK_2 message, send `Done` messages, and terminate. Parties might terminate earlier; as mentioned, if one honest party terminates, it outputs its polynomial, and we are also guaranteed that at least $t+1$ other honest parties terminate with their input polynomials. Therefore, weak validity holds. ◀

4.2 Asynchronous Data Dissemination

▶ **Definition 4.4** (Asynchronous Data Dissemination). *A protocol for parties $P_1, \ldots, P_n$ where each P_i has input $f_i(x)$ (of degree d over $\mathbb{F}$, might be $\perp$) is an* asynchronous data dissemination protocol tolerating t malicious parties *if it satisfies the following property:*

- **Agreement and termination:** *If there exists a polynomial $f(x)$ such that at least $t+1$ honest parties start with the input $f(x)$, and all other honest parties start with input $\perp$, then all honest parties terminate and output $f(x)$.*

Protocol 4.5: Asynchronous Data Dissemination

Input: Each party P_i holds $f_i(x)$ as input, of degree at most d. Some P_i might have input $\perp$.
The protocol:

1. Initialize a multi-set $M_i = \emptyset$ and $S_i = \emptyset$. If $f_i(x) \neq \perp$, send to each P_j its point $(\texttt{YourPoint}, f_i(j))$.
2. **Upon** receiving $(\texttt{YourPoint}, u_j)$ from P_j, add u_j to M_i.
3. **Upon** some u_i appearing $t+1$ times in M_i, send $(\texttt{MyPoint}, u_i)$ to all parties.
4. **Upon** receiving $(\texttt{MyPoint}, u_j)$ from P_j, add (j, u_j) to S_i. **Upon** $|S_i| \geq d+t+1$ execute the following:
   a. Run $\mathsf{RSDec}(S_i)$ and try to decode a polynomial $f_i(x)$ of degree at most d that agrees with S_i on at at least $d+t+1$ values.
   b. If no such polynomial exists, then wait to receive more points in S_i and retry.
   c. If such a polynomial $f_i(x)$ is computed, set $f_i(x)$ to be the resultant value
5. **Upon** unique decoding of $f_i(x)$, terminate and output $f_i(x)$.

▶ **Theorem 4.6.** *Protocol 4.5 is an asynchronous data-dissemination protocol tolerating $t < n/3$ malicious parties. The protocol takes 2 rounds and $O(nL + n^2 \log n)$ communication, where each party starts with an input of size L.*

Proof. Assuming that $t+1$ honest parties hold the input $f(x)$ and the others hold $\perp$, we get that each honest party P_j eventually receives $f(j)$ at least $t+1$ times and therefore sends to everyone the message $(\texttt{MyInput}, f(j))$. As a result, all honest parties eventually send that message. Therefore, all honest parties will eventually succeed in their Reed Solomon decoding and terminate with the polynomial $f(x)$. ◀

4.3 Reliable Broadcast

▶ **Definition 4.7** (Reliable Broadcast). *A protocol for parties $P_1, \ldots, P_n$ where a distinguished party P^* has an input $f(x)$ of degree at most d over $\mathbb{F}$ is a* reliable broadcast tolerating t malicious parties *if the following properties hold:*

- **Validity:** *If the sender P^* is honest then all honest parties terminate and output $f(x)$.*
- **Agreement:** *If two honest parties terminate, their output is the same.*
- **Termination:** *If an honest party terminates, then all honest parties will eventually terminate.*

Protocol 4.8: Reliable Broadcast

Input: The sender holds a polynomial $f(x)$ of degree at most d.
The protocol:

1. **The sender:** Send $f(x)$ to each party P_i.
2. **Each party P_i: Upon** receiving $f_i^{(1)}(x)$ from the sender, participate in Dispersal (Protocol 4.2) with input $f_i^{(1)}(x)$.
3. **Each party P_i: Upon** receiving an output $f_i^{(2)}(x)$ from the dispersal protocol, participate in asynchronous data dissemination protocol (Protocol 4.5) with input $f_i^{(2)}(x)$.
4. **Upon** receiving an output $f_i^{(3)}(x)$ from the data dissemination protocol, terminate and output $f_i^{(3)}(x)$.

▶ **Theorem 4.9.** *Protocol 4.8 is a reliable broadcast protocol tolerating $t < n/3$ malicious parties (as per Definition 4.7). The protocol takes 6 rounds and $O(nL + n^2 \log n)$ bits for reliably broadcasting a message of size L.*

Proof. We show each one of the properties separately.

Validity. If the sender is honest and starts with the input $f(x)$, then eventually all honest parties start the Dispersal protocol with the input $f(x)$. From the weak-validity property of the dispersal protocol (see Definition 4.1), all honest parties terminate with an output which is in $\{\perp, f(x)\}$, and at least $t+1$ honest parties terminate with $f(x)$. All honest parties then eventually continue to the asynchronous data dissemination. According to that protocol, if at least $t+1$ honest parties start with the same input $f(x)$, and all other honest parties start with $\perp$, then all honest parties must terminate, and with output $f(x)$. Therefore, all honest parties eventually terminate the reliable broadcast protocol, with an output $f(x)$.

Agreement and termination. Assume that some honest party terminates. This implies that both the dispersal and the data dissemination protocols must terminate (an honest party does not participate in the data dissemination before terminating the dispersal). If the dispersal protocol terminates, then from its security properties, we must have that at least $t+1$ honest parties terminate with the same polynomial $f(x)$, and all the other honest parties terminate with an output $\perp$. The parties then input those inputs to the data-dissemination protocol, which guarantees that under this exact input – all honest parties terminate with the same output $f(x)$. We conclude that all honest parties terminate with the same output $f(x)$. ◀

Reducing one-round. As in the case of synchrony, we can reduce one round of interaction by sending the first round of the data-dissemination protocol together with the last round of the data-dispersal protocol. Specifically, each party that sends `Done` message to P_j (the last round of the weak dispersal) sends, together with it, the point $f_i(j)$ to party P_j (the first message of the data dissemination).

Making the protocol balanced. In the reliable broadcast protocol as described in Protocol 4.8, the communication complexity of the sender is $n(d+1)\log n$ bits, whereas the communication complexity of every other party is just $O(n \log n)$ bits. As in the case of synchrony, to balance the communication complexity of the sender, the sender can first send to each party P_i its point $f(i)$. The parties then send it to all other parties and execute Reed Solomon decoding repeatedly to reconstruct the polynomial $f(x)$. Specifically, each party waits to receive at least $d+t+1$ points and executes Reed Solomon decoding to recover a polynomial that agrees with at least $d+t+1$ points.

References

1 Ittai Abraham and Gilad Asharov. Gradecast in synchrony and reliable broadcast in asynchrony with optimal resilience, efficiency, and unconditional security. In *PODC '22: ACM Symposium on Principles of Distributed Computing, 2022*, pages 392–398. ACM, 2022. doi:10.1145/3519270.3538451.

2 Ittai Abraham, Gilad Asharov, Shravani Patil, and Arpita Patra. Asymptotically free broadcast in constant expected time via packed VSS. In *Theory of Cryptography - 20th International Conference, TCC 2022*, volume 13747 of *Lecture Notes in Computer Science*, pages 384–414. Springer, 2022. doi:10.1007/978-3-031-22318-1_14.

3 Ittai Abraham, T.-H. Hubert Chan, Danny Dolev, Kartik Nayak, Rafael Pass, Ling Ren, and Elaine Shi. Communication complexity of byzantine agreement, revisited. *Distributed Comput.*, 36(1):3–28, 2023. doi:10.1007/S00446-022-00428-8.

4 Ittai Abraham, Philipp Jovanovic, Mary Maller, Sarah Meiklejohn, Gilad Stern, and Alin Tomescu. Reaching consensus for asynchronous distributed key generation. In *PODC '21: ACM Symposium on Principles of Distributed Computing, 2021*, pages 363–373. ACM, 2021. doi:10.1145/3465084.3467914.

5 Ittai Abraham and Andrew Lewis-Pye. Phase-king through the lens of gradecast: A simple unauthenticated synchronous byzantine agreement protocol. *Decentralized Thoughts, Blog Post*, 2022. URL: https://decentralizedthoughts.github.io/2022-06-09-phase-king-via-gradecast/.

6 Nicolas Alhaddad, Sourav Das, Sisi Duan, Ling Ren, Mayank Varia, Zhuolun Xiang, and Haibin Zhang. Balanced byzantine reliable broadcast with near-optimal communication and improved computation. In *PODC '22: ACM Symposium on Principles of Distributed Computing*, pages 399–417. ACM, 2022. doi:10.1145/3519270.3538475.

7 Gilad Asharov and Anirudh Chandramouli. Perfect (parallel) broadcast in constant expected rounds via statistical VSS. In *Advances in Cryptology - EUROCRYPT 2024*, volume 14655 of *Lecture Notes in Computer Science*, pages 310–339. Springer, 2024. doi:10.1007/978-3-031-58740-5_11.

8 Michael Ben-Or. Another advantage of free choice: Completely asynchronous agreement protocols (extended abstract). In *Proc. of the Annual Symposium on Principles of Distributed Computing (PODC)*, 1983.

9 Michael Ben-Or, Danny Dolev, and Ezra N. Hoch. Brief announcement: Simple gradecast based algorithms. In *Distributed Computing, 24th International Symposium, DISC 2010.*, volume 6343 of *Lecture Notes in Computer Science*, pages 194–197. Springer, 2010. doi:10.1007/978-3-642-15763-9_18.

10 Piotr Berman, Juan A Garay, and Kenneth J Perry. Bit optimal distributed consensus. In *Computer science*. 1992.

11 Gabriel Bracha. Asynchronous byzantine agreement protocols. *Inf. Comput.*, 75(2):130–143, 1987. doi:10.1016/0890-5401(87)90054-X.

12 Christian Cachin and Stefano Tessaro. Asynchronous veri.able information dispersal. In *24th IEEE Symposium on Reliable Distributed Systems (SRDS 2005), 2005*, pages 191–202. IEEE Computer Society, 2005. doi:10.1109/RELDIS.2005.9.

13 Jinyuan Chen. Optimal error-free multi-valued byzantine agreement. In *DISC 2021*, volume 209 of *LIPIcs*, pages 17:1–17:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.DISC.2021.17.

14 Sourav Das, Zhuolun Xiang, and Ling Ren. Asynchronous data dissemination and its applications. In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2705–2721. ACM, 2021. doi:10.1145/3460120.3484808.

15 Sourav Das, Zhuolun Xiang, and Ling Ren. Balanced quadratic reliable broadcast and improved asynchronous verifiable information dispersal. *IACR Cryptol. ePrint Arch.*, page 52, 2022. URL: https://eprint.iacr.org/2022/052.

16 Danny Dolev and Rüdiger Reischuk. Bounds on information exchange for byzantine agreement. In *ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, 1982*, pages 132–140. ACM, 1982. doi:10.1145/800220.806690.

17 Paul Feldman and Silvio Micali. Optimal algorithms for byzantine agreement. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 148–161. ACM, 1988. doi:10.1145/62212.62225.

18 Michael J. Fischer and Nancy A. Lynch. A lower bound for the time to assure interactive consistency. *Information Processing Letters*, 1982.

19 Chaya Ganesh and Arpita Patra. Optimal extension protocols for byzantine broadcast and agreement. *Distributed Comput.*, 34(1):59–77, 2021. doi:10.1007/S00446-020-00384-1.

20 Jonathan Katz and Chiu-Yuen Koo. On expected constant-round protocols for byzantine agreement. In *Annual International Cryptology Conference*, 2006.

21 Jonathan Katz and Chiu-Yuen Koo. On expected constant-round protocols for byzantine agreement. *J. Comput. Syst. Sci.*, 75(2):91–112, 2009. doi:10.1016/J.JCSS.2008.08.001.

22 Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982. doi:10.1145/357172.357176.

23 Guanfeng Liang and Nitin H. Vaidya. Error-free multi-valued consensus with byzantine failures. In *Proceedings of the 30th Annual ACM Symposium on Principles of Distributed Computing, PODC 2011*, pages 11–20. ACM, 2011. doi:10.1145/1993806.1993809.

24 Andrew D. Loveless, Ronald G. Dreslinski, and Baris Kasikci. Optimal and error-free multi-valued byzantine consensus through parallel execution. *IACR Cryptol. ePrint Arch.*, page 322, 2020. URL: https://eprint.iacr.org/2020/322.

25 Kartik Nayak, Ling Ren, Elaine Shi, Nitin H. Vaidya, and Zhuolun Xiang. Improved extension protocols for byzantine broadcast and agreement. In *34th International Symposium on Distributed Computing, DISC 2020*, volume 179 of *LIPIcs*, pages 28:1–28:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.DISC.2020.28.

26 Arpita Patra. Error-free multi-valued broadcast and byzantine agreement with optimal communication complexity. In *Principles of Distributed Systems - 15th International Conference, OPODIS 2011*, volume 7109 of *Lecture Notes in Computer Science*, pages 34–49. Springer, 2011. doi:10.1007/978-3-642-25873-2_4.

27 Marshall C. Pease, Robert E. Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, 1980. doi:10.1145/322186.322188.

28 M. O. Rabin. Randomized byzantine generals. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 1983.

29 Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980. doi:10.1145/322217.322225.

30 Jianjun Zhu, Fan Li, and Jinyuan Chen. Communication-efficient and error-free gradecast with optimal resilience. In *2023 IEEE International Symposium on Information Theory (ISIT)*, pages 108–113, 2023. doi:10.1109/ISIT54713.2023.10206579.

31 Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation*, volume 72 of *Lecture Notes in Computer Science*, pages 216–226. Springer, 1979. doi:10.1007/3-540-09519-5_73.

Improved Lower Bounds for 3-Query Matching Vector Codes

Divesh Aggarwal
Centre for Quantum Technologies, National University of Singapore, Singapore

Pranjal Dutta
National University of Singapore, Singapore

Zeyong Li
Centre for Quantum Technologies, National University of Singapore, Singapore

Maciej Obremski
National University of Singapore, Singapore

Sidhant Saraogi
Georgetown University, Washington, DC, USA

Abstract

A Matching Vector (**MV**) family modulo a positive integer $m \geq 2$ is a pair of ordered lists $\mathcal{U} = (\boldsymbol{u}_1, \cdots, \boldsymbol{u}_K)$ and $\mathcal{V} = (\boldsymbol{v}_1, \cdots, \boldsymbol{v}_K)$ where $\boldsymbol{u}_i, \boldsymbol{v}_j \in \mathbb{Z}_m^n$ with the following property: for any $i \in [K]$, the inner product $\langle \boldsymbol{u}_i, \boldsymbol{v}_i \rangle = 0 \pmod m$, and for any $i \neq j$, $\langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle \neq 0 \pmod m$. An **MV** family is called *r-restricted* if inner products $\langle u_i, v_j \rangle$, for all i, j, take *at most* r different values. The r-restricted **MV** families are extremely important since the only known construction of constant-query subexponential locally decodable codes (LDCs) are based on them. Such LDCs constructed via matching vector families are called *matching vector codes*. Let $\mathbf{MV}(m, n)$ (respectively $\mathbf{MV}(m, n, r)$) denote the *largest* K such that there exists an **MV** family (respectively r-restricted **MV** family) of size K in $\mathbb{Z}_m^n$. Such a **MV** family can be transformed in a black-box manner to a *good* r-query locally decodable code taking messages of length K to codewords of length $N = m^n$.

For small prime m, an *almost* tight bound $\mathbf{MV}(m, n) \leq O(m^{n/2})$ was first shown by Dvir, Gopalan, Yekhanin (FOCS'10, SICOMP'11), while for general m, the same paper established an upper bound of $O(m^{n-1+o_m(1)})$, with $o_m(1)$ denoting a function that goes to zero when m grows. For any arbitrary constant $r \geq 3$ and composite m, the best upper bound till date on $\mathbf{MV}(m, n, r)$ is $O(m^{n/2})$, is due to Bhowmick, Dvir and Lovett (STOC'13, SICOMP'14).In a breakthrough work, Alrabiah, Guruswami, Kothari and Manohar (STOC'23) implicitly improve this bound for 3-restricted families to $\mathbf{MV}(m, n, 3) \leq O(m^{n/3})$.

In this work, we present an upper bound for $r = 3$ where $\mathbf{MV}(m, n, 3) \leq m^{n/6+O(\log n)}$, and as a result, any 3-query matching vector code must have codeword length of $N \geq K^{6-o(1)}$.

2012 ACM Subject Classification Theory of computation → Error-correcting codes

Keywords and phrases Locally Decodable Codes, Matching Vector Families

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.2

Related Version *Previous Version*: https://eccc.weizmann.ac.il/report/2024/061/

Funding *Divesh Aggarwal*: Funded by the bridging grant at Centre for Quantum Technologies titled "Quantum algorithms, complexity, and communication".
Pranjal Dutta: Funded by NRF Singapore under the project "Computational Hardness of Lattice Problems and Implications" [NRF-NRFI09-0005].
Zeyong Li: Funded by NRF Singapore under the project "Computational Hardness of Lattice Problems and Implications" [NRF-NRFI09-0005].
Maciej Obremski: Funded by MOE tier 2 grant T2EP20223-0024, Breaking the Box – On Security of Cryptographic Devices.
Sidhant Saraogi: Funded in part by the National Science Foundation CAREER award to Justin Thaler (grant CCF-1845125).



© Divesh Aggarwal, Pranjal Dutta, Zeyong Li, Maciej Obremski, and Sidhant Saraogi;
licensed under Creative Commons License CC-BY 4.0



Acknowledgements The authors would like to thank Sivakanth Gopi and Anup Rao for sharing a proof of Theorem 21. They would also like to thanks anonymous reviewers for their helpful comments.

1 Introduction

1.1 Locally Decodable Codes

A code $C : \Sigma^K \to \Sigma^N$ is said to be (r, δ, ε)-locally decodable if for every $i \in [K]$, the i-th coordinate of the message $\boldsymbol{x}_i \in \Sigma$ can be recovered with probability at least $1 - \varepsilon$ by a randomized decoding procedure that makes only r queries to the codeword, i.e., reads the codeword at r positions even if the codeword is corrupted on up to δN locations.

Locally decodable codes were formally introduced in [19] but had already been studied informally in the context of the PCP theorem [3, 4]. Locally decodable codes have found applications in many areas of computer science such as worst-case to average-case reductions, private information retrieval, secure multi-party computation, derandomization, matrix rigidity, data structures, and fault-tolerant computation. See [25] for a detailed survey.

A central research question is to understand the largest possible message length K (as a function of N) that can be achieved by a r-query locally decodable code. For the simplest non-trivial setting of $r = 2$, we have a nearly complete understanding – the Hadamard code achieves $K = \log_2 N$, and it was shown in [20, 12] that this is the best possible up to a constant factor, i.e., $K = O(\log_2 N)$.

Much less is understood about the dependence between K and N for the number of queries $r \geq 3$. In particular, the only known constructions for r-query locally decodable codes for a constant $r \geq 3$ are based on families of matching vector codes [24, 10, 9]. These constructions achieve $K = (\log N)^{\Omega(\log \log N)}$. On the other hand, the known upper bounds on K are very far from what is achieved by these constructions. In a series of works [19, 20, 22, 23, 6], it has been shown that $K = N^{1-2/(r+1)} \operatorname{polylog} N$, when r is odd, and $K = N^{1-2/r} \operatorname{polylog} N$, when r is even. For the special case where $r = 3$, it was shown that $K = O(N^{1/3} \log^2 N)$ in a celebrated recent result [2].

1.2 Matching Vector Families and Application to Locally Decodable Codes

A matching vector (**MV**) family is a combinatorial object that has been used in different contexts such as Ramsey graphs and weak representation of OR polynomials[11, 17, 1, 13], but the most prominent application of matching vector families is in the construction of constant-query locally decodable codes [9, 25]. A matching vector family is defined by two ordered lists $\mathcal{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K)$ and $\mathcal{V} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K)$ where for all $i \in [K]$, $\boldsymbol{u}_i, \boldsymbol{v}_i$ are in $\mathbb{Z}_m^n$ for some positive integers $m, n > 1$. An S-matching vector family can be defined as follows.

▶ **Definition 1** (Matching Vector Family). *Let $S \subseteq \mathbb{Z}_m \setminus \{0\}$. An S-matching vector family is a set of vectors $\mathcal{U} = \{u_1, \ldots, u_K\}$, $\mathcal{V} = \{v_1, \ldots, v_K\}$, $\mathcal{U}, \mathcal{V} \subseteq \mathbb{Z}_m^n$ such that:*

- $\forall i \in [K], \langle u_i, v_i \rangle = 0$.
- $\forall i \neq j,\ i, j \in [K], \langle u_i, v_j \rangle \in S$.

An S-matching vector family is often called $(|S| + 1)$-restricted matching vector family, and such a family gives a locally decodable code for messages of block length K, codeword length $N = m^n$, and number of queries $r = |S|+1$. See Theorem 15 ([25]) for the construction of a locally decodable code from a matching vector family.

Finally, we denote $\mathbf{MV}(m, n)$ (respectively $\mathbf{MV}(m, n, r)$), as the *largest* K such that there exists an **MV** family (respectively r-restricted **MV** family) of size K in $\mathbb{Z}_m^n$.

1.3 Upper Bounds for Matching Vector Families

Since the only known approach that has led to constructions of constant-query locally decodable codes is via constructing a large matching vector family, it is natural to ask what is the largest K as a function of m, n for which there is a r-restricted matching vector family of size K. This in particular will give an upper bound for the rate of a r-query locally decodable code via matching vector codes, i.e., any r-query locally decodable code achieving a better rate must be via some novel approach that does not require a matching vector family.

This problem was first studied in [9], where the authors showed that when m is a large prime, $\mathbf{MV}(m, n) \leq O(m^{n/2})$. On the other hand, the same paper established a general upper bound of $O(m^{n-1+o_m(1)})$, with $o_m(1)$ denoting a function that goes to zero when a composite m grows. In [8], the authors improved this bound to $\text{poly}(m) \cdot m^{.625n}$, for a very special case when $m = pq$, for distinct primes p and q such that $p \approx q$, and $n \ll m$. On the other hand, when $m \ll n$ is a prime, it can be shown via linear algebraic methods that $\mathbf{MV}(m, n) \leq O(n^{m-1})$ [5].

Bhowmick, Dvir and Lovett [7] showed how to extend the results of [9] to the case of m being a composite integer. They showed that $\mathbf{MV}(m, n, r) \leq r^{O(r \log r)} m^{n/2}$, i.e., for constant query codes, they showed that $K = O(\sqrt{N})$.

Additionally, in the same paper, Bhowmick, Dvir and Lovett also showed that under the (recently proved) polynomial Freiman Ruzsa (PFR) theorem [16], for m constant, $\mathbf{MV}(m, n) \leq m^{O(n/\log n)}$.

1.4 Our Result

The main contribution of the paper is to give a better upper bound for 3-restricted matching vector families.

▶ **Theorem 2.** *Let $n, m \geq 2$ be positive integers with $n > 10m^3$. Suppose m has atleast two distinct prime factors, then*

$$\mathbf{MV}(m, n, 3) \leq m^{\frac{n+2\log(n+1)}{2q}+2m^2},$$

where q is the second-smallest *prime dividing m.*

We already know a polynomial bound for prime powers due to [15] which we state below and prove for completeness in Section 2.4.

▶ **Theorem 3.** *Suppose $n, m \geq 2$ be positive integers where m is a constant prime power. Then $\mathbf{MV}(m, n, 3) \leq (2emn)^{m-1}$.*

This allows us to conclude an upper bound for arbitrary m:

▶ Remark 4. When m has at least one factor which is not 2, that immediately gives a $m^{n/6+O(\log n)}$ upper bound, while if the constant m is a power of 2, the bound is just $\text{poly}(n)$, as mentioned above.

Finally, we can also derive a lower bound for the 3-query LDCs derived from matching vector families:

▶ **Corollary 5.** *Suppose $C : \Sigma^K \to \Sigma^N$ is a 3-query locally decodable code constructed from a 3-restricted matching vector family in a black-box manner (as in Theorem 15), then $N \geq K^{6-o_K(1)}$ where $o_K(1)$ vanishes as K increases.*

1.5 Concurrent Work

Concurrent to our work, the Polynomial Freiman Ruzsa conjecture was proven by Gowers, Green, Manners and Tao [16] for all abelian groups with bounded torsion! As a corollary (via Theorem 6.4 of [7]), one could conclude that

$$\mathbf{MV}(m,n) \leq 2^{1200c(m)m^{6\log m}n/\log n},$$

where $c(m)$ is some explicit constant depending only on m. Infact, if we plug in the bound of [16] to Lemma 6.3 from [7], we can conclude that $c(m) \geq m^3$. Note that this bound is non-trivial (i.e., less than m^n) only when $n \geq 2^{\frac{1200\cdot 6^{3+6\log 6}}{\log 6}} > 2^{2^{56}}$ (plugging in $m = 6$) and beats our bound only when $n \geq 2^{2^{58}}$.

While this result supersedes ours in many parameter regimes, the techniques and approaches are very different. And our approach might have the potential of breaking the $2^{n/\log n}$ barrier. (See discussions below.)

1.6 Our Techniques

In the discussion below, we want to prove upper bounds for a matching vector family $\mathcal{U}, \mathcal{V}$ of size K over $\mathbb{Z}_m^n$ for some positive integers m and n.

Techniques from [7]. We begin by describing Bhowmick, Dvir, and Lovett's ([7]) approach towards getting an upper bound of $m^{n/2}$. They observe that if we consider the random variables $\boldsymbol{U}, \boldsymbol{V}$ as sampled independently and randomly from $\mathcal{U} = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K\}$ and $\mathcal{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K\}$, respectively, then we have that $\Pr[\langle \boldsymbol{U}, \boldsymbol{V}\rangle = 0] = \frac{1}{K}$, and hence $\langle \boldsymbol{U}, \boldsymbol{V}\rangle \bmod m$ is *far* from uniform.

If m is a prime, then standard properties of the inner product two-source extractor imply that $\mathbf{H}_\infty(\boldsymbol{U}) + \mathbf{H}_\infty(\boldsymbol{V}) = 2\log K$ is not much bigger than $n \log m$, else the inner product extractor's output would be very close to uniform. This straightaway yields the desired bound of $K \lessapprox m^{n/2}$.

For composite m, this approach does not work directly, since $\mathbb{Z}_m$ is *not* a field. In particular, say for $m = 6$, it is possible to have two random variables distributed uniformly over $\{0, 2, 4\}^n$ such that their inner product is *far* from uniform modulo 6, but the sum of the min-entropies is $2n \log_2 3 \gg n \log 6$. The authors overcome this by showing that this is essentially the *only* bad case, i.e., it can be attributed to some "bad" factor $s \mid m$ (in the example above, $s = 2$). Hence, with some careful inductive argument on the modulus m/s, one can still force out the fundamental observation: $\mathbf{H}_\infty(\boldsymbol{U}) + \mathbf{H}_\infty(\boldsymbol{V}) = 2\log K$, which cannot be too large due to the biased behavior of $\langle \boldsymbol{U}, \boldsymbol{V}\rangle$, showing the desired upper bound on K.

Going beyond $m^{n/2}$. A natural attempt towards improving the upper bound for K would be to consider the inner product $\langle \boldsymbol{U}^\dagger, \boldsymbol{V}^\dagger\rangle$ where $\boldsymbol{U}^\dagger = \boldsymbol{U}_1 + \cdots + \boldsymbol{U}_\ell$ for some integer ℓ with each $\boldsymbol{U}_i$ sampled uniformly and independently from $\mathcal{U}$ (and $\boldsymbol{V}^\dagger$ similarly defined), and hope for two nice events:

(a) $\mathbf{H}_\infty(\boldsymbol{U}^\dagger) + \mathbf{H}_\infty(\boldsymbol{V}^\dagger) \approx 2\ell \log K \gg 2\log K$;

(b) $\langle \boldsymbol{U}^\dagger, \boldsymbol{V}^\dagger\rangle$ remains far from uniform just like $\langle \boldsymbol{U}, \boldsymbol{V}\rangle$. And hence $\mathbf{H}_\infty(\boldsymbol{U}^\dagger) + \mathbf{H}_\infty(\boldsymbol{V}^\dagger) \lessapprox m \log n$.

In other words, we are trying to boost the lower bound on min-entropy by adding ℓ independent samples of $\boldsymbol{U}$, and at the same time maintain the upper bound on min-entropy. And if the above is true, one would obtain a substantial improvement on the upper bound (e.g. $K \leq m^{n/4}$ even for the smallest non-trivial $\ell = 2$).

In the rest of this proof overview, we will walk through how we show the two events above are true for any $\{\alpha, \beta\}$-matching vector family. For simplicity, let $m = pq$ for distinct primes p and q, with $q > p$. And one could further assume that $\alpha = 0 \pmod p, \alpha \neq 0 \pmod q$, and $\beta = 0 \pmod q, \beta \neq 0 \pmod p$; see Lemma 25 for more details.

Event (a)

In order to show that $\mathbf{H}_\infty(\boldsymbol{U}^\dagger) \approx \ell \log K$, it is sufficient to show that $\boldsymbol{U}^\dagger = \sum_{i=1}^{\ell} \boldsymbol{U}_i$ *does not* generate too many collisions.

Omitting most technical details, we show that if $\boldsymbol{u}_1 + \ldots + \boldsymbol{u}_\ell = \boldsymbol{u}'_1 + \ldots + \boldsymbol{u}'_\ell$ (i.e. a collision), then it must be the case that $\langle \boldsymbol{u}'_i, \boldsymbol{v}_1 \rangle = \alpha$ for all $1 \leq i \leq \ell \leq q$, where $\boldsymbol{v}_1$ is the matching vector for $\boldsymbol{u}_1$. This is due to the heavy constraint that – (1) any inner product can only take values from $\{0, \alpha, \beta\}$, and (2) inner product with $\boldsymbol{v}_1$ gives an expression of the form $x_1\alpha + x_2\beta$, on the LHS, while $y_1\alpha + y_2\beta$, on the RHS, for nonnegative integers x_i, y_i such that $x_1 + x_2 \leq \ell - 1$. and $y_1 + y_2 \leq \ell$. (Remember $\langle \boldsymbol{u}_1, \boldsymbol{v}_1 \rangle = 0$.)

This anomaly (that there is plausibly 1 less term in the LHS) is sufficient to conclude that if there are too many collisions, one can easily extract a large $\{\alpha\}$-matching vector family, which would contradict a known linear upper bound, since a simple pigeon-hole principle argument shows that $\mathbf{MV}(pq, n, 2) \leq n + 1$; see Lemma 18.

In the actual proof, for technical reasons, we work with a slightly different distribution $\boldsymbol{U}^\star$, which adds up ℓ *distinct* random vectors from $\mathcal{U}$. In other words, one may view $\boldsymbol{U}^\dagger$ as sampling *with replacement* and $\boldsymbol{U}^\star$ as sampling *without replacement* from $\mathcal{U}$. These two distributions are so close to each other that we are able to substitute one with the other without incurring much loss in the final argument. Moreover, this distinction also helps us to argue that the sum of the coefficients in LHS is *exactly* $\ell - 1$, while it is *exactly* ℓ in RHS. For the detailed proof on lower bounding the min-entropy, see Section 3.2.

Event (b)

To show that $\langle \boldsymbol{U}^\dagger, \boldsymbol{V}^\dagger \rangle$ is far from uniform, we directly examine its bias $\left|\mathbb{E}[\omega^{\langle \boldsymbol{U}^\dagger, \boldsymbol{V}^\dagger \rangle}]\right|$ in terms of Fourier coefficients (where ω is a primitive m-th root of unity). Via standard Fourier-analytic techniques, and using the "almost" multiplicative property of the bias (Lemma 11), we can *almost* show that

$$\left|\mathbb{E}[\omega^{\langle \boldsymbol{U}^\dagger, \boldsymbol{V}^\dagger \rangle}]\right| \gtrapprox \left|\mathbb{E}[\omega^{\langle \boldsymbol{U}, \boldsymbol{V} \rangle}]\right|^{\ell^2} .$$

That is, adding ℓ independent samples of $\boldsymbol{U}$ (and $\boldsymbol{V}$) will *not* smooth out the inner product for not too large ℓ (see Lemmas 11 and 13).

On the other hand, since $\langle \boldsymbol{U}, \boldsymbol{V} \rangle$ takes only 3 possible values with 0 showing up with negligible probability $\frac{1}{K}$, we can show that $\left|\mathbb{E}[\omega^{\langle \boldsymbol{U}, \boldsymbol{V} \rangle}]\right|$ has to be large! And this is essentially all we need. For the detailed proof on upper bounding the min-entropy, see Section 3.1.

Combining all of the above and choosing $\ell = q$, we get $2q \log K \lessapprox \mathbf{H}_\infty(\boldsymbol{U}^\star) + \mathbf{H}_\infty(\boldsymbol{V}^\star) \lessapprox n \log m$ which gives us the desired upper bound of $K \lessapprox m^{n/2q}$. For a general m one can always reduce it to working with $m = p^s q^t$, for primes p and q; see Lemmas 24 and 25. And for $m = p^s q^t$, the argument is similar to the above.

Lastly, it is worth noting that via a more direct analysis on the min-entropy of carefully chosen random variables, we are able to avoid the inductive argument in [7], which was tailored for composite m.

1.7 Conclusions and Open Questions

Our approach can be broadly summarized as considering $\boldsymbol{U}^\dagger := \boldsymbol{U}_1 + \cdots + \boldsymbol{U}_\ell$, and similarly $\boldsymbol{V}^\dagger := \boldsymbol{V}_1 + \cdots + \boldsymbol{V}_\ell$, and then proving that

1. $\omega^{\langle \boldsymbol{U}^\dagger, \boldsymbol{V}^\dagger \rangle}$ is far from uniform, where ω is the m-th root of unity, and
2. $\mathbf{H}_\infty(\boldsymbol{U}^\dagger) \approx \ell \cdot \mathbf{H}_\infty(\boldsymbol{U})$ for $\ell = q$.

The first item implies that $\mathbf{H}_\infty(\boldsymbol{U}^\dagger) + \mathbf{H}_\infty(\boldsymbol{V}^\dagger)$ is not much larger than $m \log n$, and this combined with the second item implies that $\mathbf{H}_\infty(\boldsymbol{U}) = \mathbf{H}_\infty(\boldsymbol{V})$ is small, giving an upper bound on K.

We prove these two items for $\ell = q$, and for r-restricted families for $r = 3$. Notice that the approach is much more general, and it is just a shortcoming of our proof techniques that doesn't allow us to go beyond $\ell > q$, or $r > 3$. In particular, for $r = 3$, the statement (1) above holds for $\ell \gg q$, and we leave it as an open question whether one can prove that for $\ell \gg q$, $\mathbf{H}_\infty(\boldsymbol{U}^\dagger) \gg q\mathbf{H}_\infty(\boldsymbol{U})$. This will immediately imply a better upper bound on K for 3-query matching vector codes.

Similarly, we leave it as an open question whether the same approach can be extended to a larger number of queries.

2 Preliminaries

2.1 Random Variable and Entropy

▶ **Definition 6** (Min-Entropy). *Let X be a random variable over a set $\mathcal{X}$, the Min-Entropy is defined as:*

$$\mathbf{H}_\infty(X) = \min_{x \in \mathcal{X}} -\log(\Pr[X = x]) \ .$$

▶ **Definition 7** (Collision Entropy). *Let X and Y be independent and identically distributed random variables, the Collision Entropy is defined as:*

$$\mathbf{H}_2(X) = -\log \Pr[X = Y] \ .$$

Alternatively,

$$\mathbf{H}_2(X) = -\log \sum_x (\Pr[X = x])^2 .$$

▶ **Observation 8** (Relation Between Collision and Min-Entropy). *For any random variable X, it holds that*

$$\mathbf{H}_2(X) \geq \mathbf{H}_\infty(X) \ .$$

▶ **Lemma 9.** *Let X and Y be independent random variables of the same domain $\mathcal{X}$ closed under addition. It holds that*

$$\mathbf{H}_\infty(X) \leq \mathbf{H}_\infty(X + Y) \ .$$

Proof. For any $z \in \mathcal{X}$, we have:

$$\begin{aligned}
\Pr[X + Y = z] &= \sum_{y \in \mathcal{X}} \Pr[X = z - y | Y = y] \Pr[Y = y] \\
&= \sum_{y \in \mathcal{X}} \Pr[X = z - y] \Pr[Y = y] \\
&\leq \sum_{y \in \mathcal{X}} (\max_x \Pr[X = x]) \Pr[Y = y] \\
&\leq \max_x \Pr[X = x] \ .
\end{aligned}$$

Therefore,

$$\mathbf{H}_\infty(X) \;=\; \log \frac{1}{\max_x \Pr[X=x]} \;\leq\; \log \frac{1}{\max_z \Pr[X+Y=z]} \;=\; \mathbf{H}_\infty(X+Y)\;,$$

as desired. ◀

Let ω be a primitive root of unity of order m. Let X and Y be two probability distributions over $\mathbb{Z}_m^n$. The bias of the output distribution wrt ω, denoted $\mathbb{E}[\omega^{\langle X,Y\rangle}]$, can be defined as follows:

$$\mathbb{E}[\omega^{\langle X,Y\rangle}] \;=\; \mathbb{E}_{x\sim X, y\sim Y}[\omega^{\langle x,y\rangle}] \;=\; \sum_{x,y} X(x)Y(y)\omega^{\langle x,y\rangle}\;.$$

▶ **Lemma 10** (Lemma 2.14 from [7]). *Let ω be a primitive root of unity of order m. Let X and Y be two probability distributions over $\mathbb{Z}_m^n$. If $|\mathbb{E}[\omega^{\langle X,Y\rangle}]| \geq \epsilon$, then $\mathbf{H}_2(X)+\mathbf{H}_2(Y) \leq n\log m + 2\log(1/\epsilon)$.*

▶ **Lemma 11** (Lemma 3.3 from [21]). *Let ω be a primitive root of unity of order m. Let X and Y be two probability distributions over $\mathbb{Z}_m^n$, then $|\mathbb{E}[\omega^{\langle X-X,Y\rangle}]| \geq |\mathbb{E}[\omega^{\langle X,Y\rangle}]|^2$. Here $X-X$ is shorthand for taking the difference of two samples drawn independently according to the distribution X.*

▶ Remark 12. Lemma 11 was stated for distributions over $\mathbb{F}^n$ for any finite field $\mathbb{F}$ in [21], but the proof trivially extends to distributions over $\mathbb{Z}_m^n$. For completeness, we reproduce the proof from [21] in the appendix.

▶ **Lemma 13** ([21]). *Let ω be a primitive root of unity of order m. Let X and Y be two probability distributions over $\mathbb{Z}_m^n$. For any integer $x, y \geq 0$, $|\mathbb{E}[\omega^{\langle 2^x\cdot X - 2^x\cdot X, 2^y\cdot Y - 2^y\cdot Y\rangle}]| \geq |\mathbb{E}[\omega^{\langle X,Y\rangle}]|^{2^{x+y+2}}$. Here $2^x\cdot X$ represents the distribution of adding 2^x independent samples of X.*

Proof. Notice that $(X-X)-(X-X) = 2X-2X$. Hence this follows from applying Lemma 11 $(x+1)+(y+1) = x+y+2$ times. ◀

2.2 Matching Vector Families and Codes

Much of these definitions follow the same notation as Yekhanin's survey [25].

▶ **Definition 14** (Locally Decodable Codes). *A q-ary code $C : \Sigma^K \to \Sigma^N$ with $|\Sigma| = q$ is (r,δ,ε)-locally decodable if there exists a randomized decoding algorithm $\mathcal{A}$ such that*

1. *For all $x \in \Sigma^K$, $i \in [K]$ and all $y \in \Sigma^N$ such that $\Delta(C(x), y) \leq \delta$:*

$$\Pr_{\mathcal{A}}[\mathcal{A}^y(i) = x(i)] \geq 1-\varepsilon$$

2. *$\mathcal{A}$ makes atmost r queries to y.*

The best-known constructions of locally decodable codes are derived from constructions of matching vector families over $\mathbb{Z}_m$ for composite m's. The following theorem demonstrates the parameters of locally decodable codes obtained from given parameters of a matching vector family.

▶ **Theorem 15** ([25]). *Let $\mathcal{U}, \mathcal{V}$ be a S-matching vector family over $\mathbb{Z}_m^n$ with $|\mathcal{U}| = |\mathcal{V}| = K$. Suppose $m \mid q-1$ for some prime power q, then there exists a linear code $C : \mathbb{F}_q^K \to \mathbb{F}_q^{m^n}$ that is $(|S|+1, \delta, (|S|+1)\delta)$-locally decodable for every δ.*

In the low-query complexity regime, $r = O(1)$, the best-known matching vector families are based on Grolmusz's construction of set systems.

▶ **Theorem 16** ([25, Lemma 4.8]). *Let $m = \prod_{i=1}^{t} p_i$ be a product of distinct primes. Let w be a positive integer. Let $\{e_i\}_{i=1}^{t}$ be integers such that $p_i^{e_i} \geq w^{1/t}$ for all i. Let $d = \max_i p_i^{e_i}$ and $h \geq w$ be arbitrary. Then there exists an $\binom{h}{w}$-sized σm-bounded family of matching vectors in $\mathbb{Z}_m^n$ where $n = \binom{h}{\leq d}$ and σ is an arbitrary real larger than $\sum_{i\in[t]} \frac{1}{p_i}$.*

Then, using Theorem 15, this matching vector family construction can be transformed into a linear binary LDC with the following parameters:

▶ **Theorem 17** ([18], as stated in [25], Theorem 3.11). *For every integer $t \geq 2$ and for all $K \geq 2$, there exists an $r = 3 \cdot 2^{t-2}$-query (binary) linear locally decodable code over $\mathbb{F}_{2^t}$ encoding K-long messages to $2^{t(\log K)^{1/t}(\log\log K)^{1-1/t}}$-long codewords and allowing $\delta = O(1/r)$ fraction of errors.*

In particular, when $r = 3$, there exists a code with codeword length $N = 2^{2^{\widetilde{O}(\sqrt{\log K})}}$

2.3 Upper Bounds for 2-restricted Matching Vector Families

In this section, we prove an upper bound on any $\{\alpha\}$-matching vector family over $\mathbb{Z}_m^n$. Without loss of generality, one can assume that m is a prime power p^s, because otherwise if $m = p^s q^t$, then without loss of generality $\alpha \neq 0 \pmod{p^s}$, and therefore the same set of vectors form a matching vector family over $\mathbb{Z}_{p^s}^n$.

▶ **Lemma 18.** *For any prime p and positive integer s, $\mathbf{MV}(p^s, n, 2) \leq ns + 1$.*

Proof. Suppose $\alpha \neq 0 \pmod{p^s}$. Assume towards contradiction that $\mathcal{U}, \mathcal{V}$ is a $\{\alpha\}$-matching vector family over $\mathbb{Z}_{p^s}^n$ with $|\mathcal{U}| = |\mathcal{V}| = K \geq ns + 2$.

Define $\boldsymbol{u}_i' := \boldsymbol{u}_{ns+2} - \boldsymbol{u}_i$ for $\boldsymbol{u}_i \in \mathcal{U}$. By definition of the $\{\alpha\}$-matching vector family, for $i, j \leq ns + 1$ we have:

$$\langle \boldsymbol{u}_i', \boldsymbol{v}_j \rangle = \begin{cases} \alpha, & i = j \\ 0, & i \neq j \end{cases}.$$

We claim that there exist integer coefficients $c_1, \ldots, c_{ns+1}$, not all zero, such that $-(p-1) \leq c_i \leq p - 1$, and

$$\sum_{i=1}^{ns+1} c_i \boldsymbol{u}_i' = 0 \,.$$

To see this, consider all possible sums $\sum_{i=1}^{ns+1} d_i \boldsymbol{u}_i' \in \mathbb{Z}_{p^s}^n$, for $d_i \in \{0, 1, \ldots, p-1\}$. There are p^{ns+1} such sums that can each take one of $(p^s)^n = p^{ns}$ values. By the pigeon-hole principle, there exist two distinct sums that are equal, i.e.,

$$\sum_{i=1}^{ns+1} d_i \boldsymbol{u}_i' = \sum_{i=1}^{ns+1} d_i' \boldsymbol{u}_i' \,,$$

for some tuples $(d_1, \ldots, d_{ns+1}) \neq (d_1', \ldots, d_{ns+1}')$. The claim follows by taking $c_i = d_i - d_i'$ for $i \in \{1, \ldots, ns + 1\}$.

We have that at least one of $c_1, \ldots, c_{ns+1}$ is non-zero. Without loss of generality, let $c_{ns+1} \neq 0$. Notice that $c_{ns+1} \in \{-(p-1), \ldots, -1\} \cup \{1, \ldots, p-1\}$, and hence has a multiplicative inverse modulo p^s. Let $c'_i = c_{ns+1}^{-1} c_i \pmod{p^s}$ for $i \in [ns+1]$. Thus, we have that

$$\boldsymbol{u}'_{ns+1} \;=\; -\sum_{i=1}^{ns} c'_i \boldsymbol{u}'_i \,.$$

Now take the inner product on both sides with $\boldsymbol{v}_{ns+1}$. Note that, $\langle \boldsymbol{u}'_{ns+1}, \boldsymbol{v}_{ns+1} \rangle = \alpha$, while

$$\left\langle \sum_{i=1}^{ns} c'_i \boldsymbol{u}'_i, \boldsymbol{v}_{ns+1} \right\rangle \;=\; \sum_{i=1}^{ns} c'_i \langle \boldsymbol{u}'_i, \boldsymbol{v}_{ns+1} \rangle \;=\; 0 \,.$$

Therefore, we obtain that $\alpha = 0$ which is a contradiction. ◀

▶ Remark 19. By the above lemma, we get $\mathbf{MV}(m, n, 2) \leq n \log m + 1$, since trivially, $s \leq \log m / \log p \leq \log m$.

2.4 Upper Bounds for Prime Powers

In this section, we provide a polynomial upper bound for matching vector families over $\mathbb{Z}_m$ where m is a prime power. The sketch was personally communicated to us by Sivakanth Gopi. To prove this, we need the following lemma from [14]. We state it without proof, but it follows from Lucas' Theorem.

▶ **Lemma 20.** *Let p be a prime and $s \in \mathbb{Z}^+$. There exists a polynomial $f(x_1, \ldots, x_n)$ of degree $p^s - 1$ such that for $x \in \{0,1\}^n$, $f(x) \neq 0 \pmod{p}$ iff $\sum_{i=1}^n x_i$ is divisible by p^s. Specifically, $f(x) = \prod_{j=1}^{s-1} \left(1 - \binom{\sum_{i=1}^n x_i}{p^j}^{p-1}\right)$.*

Here $\binom{\sum_{i=1}^n x_i}{k} = \sum_{i_1 < \ldots < i_k} x_{i_1} \ldots x_{i_k}$ is always a valid integer polynomial for $k \leq n$. Note that in [14], the lemma is technically stated with $1 - f$ as opposed to f. We are now ready to show the upper bound. Also, note that f takes only 2 values 1 and 0 $\pmod{p}$, depending on the divisibility property.

▶ **Theorem 21** (Upper bound for prime-power [15])**.** *Let $m = p^s$ for some prime p and $s \in \mathbb{Z}^+$. Then, for any matching vector family $\mathcal{U}, \mathcal{V} \subseteq \mathbb{Z}_m^n$ of size $t = |\mathcal{U}| = |\mathcal{V}|$ we have that $t \leq (2emn)^{m-1}$.*

Proof. We will first convert the matching vector family $\mathcal{U}, \mathcal{V}$ into a matching vector family consisting of vectors with *only zero or one* coordinates, we stress that the matching vector family is still considered over $\mathbb{Z}_m$ and $0, 1$ are interpreted as elements of $\mathbb{Z}_m$, not $\mathbb{F}_2$. Thereafter, we will argue by showing an upper bound on the rank of a specially constructed matrix, via the polynomial defined in Lemma 20.

Converting into a binary matching vector family. We will transform vectors in $\mathcal{U}, \mathcal{V}$ into vectors in $\{0,1\}^{n'} \subset \mathbb{Z}_m^{n'}$ for $n' = nm^2$, such that the pairwise inner-product $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ for $\boldsymbol{u} \in \mathcal{U}, \boldsymbol{v} \in \mathcal{V}$ is preserved over $\mathbb{Z}_m$. For any $\boldsymbol{u} \in \mathcal{U}$ (resp. $\boldsymbol{v} \in \mathcal{V}$), map each coordinate $\boldsymbol{u}_i$ (resp. $\boldsymbol{v}_i$) into the $m \times m$ matrix[1] $\boldsymbol{u}'_i$ (resp. $\boldsymbol{v}'_i$) with all 1's in the first $\boldsymbol{u}_i$ rows (resp. first $\boldsymbol{v}_i$ columns). It is easy to see that $\boldsymbol{u}_i \cdot \boldsymbol{v}_i = \langle \boldsymbol{u}'_i, \boldsymbol{v}'_i \rangle$ and

[1] Matrix is interpreted as a vector simply by reading entries of the matrix row by row.

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \sum_{i \in [n]} \boldsymbol{u}_i \cdot \boldsymbol{v}_i = \sum_{i \in [n]} \langle \boldsymbol{u}'_i, \boldsymbol{v}'_i \rangle = \langle \boldsymbol{u}', \boldsymbol{v}' \rangle$$

As a result, we can turn any matching vector family $\mathcal{U}, \mathcal{V} \subseteq \mathbb{Z}_m^n$ into one with only binary vectors in $\mathbb{Z}_m^{mn^2}$. For the rest of the argument, denote $\mathcal{U}, \mathcal{V} \subseteq \mathbb{Z}_m^{n'}$ to be the binary vectors obtained like above from the original $\mathcal{U}, \mathcal{V}$ where $n' = nm^2$.

Constructing a useful matrix. Suppose $\mathcal{U} = \{\boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(t)}\}, \mathcal{V} = \{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(t)}\}$ where all $\boldsymbol{u}^{(i)}, \boldsymbol{v}^{(j)} \in \{0,1\}^{n'} \subset \mathbb{Z}_m^{n'}$. Recall $m = p^s$, let $f(x)$ be such a polynomial as defined in Lemma 20. By the same lemma, $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \sum_{i=1}^{n'} \boldsymbol{u}_i \boldsymbol{v}_i = 0 \pmod m$ iff $f(\boldsymbol{u}_1 \boldsymbol{v}_1, \ldots, \boldsymbol{u}_{n'} \boldsymbol{v}_{n'}) \neq 0 \pmod p$. Now, we define a matrix $M \in \mathbb{Z}_p^{t \times t}$ whose rows are indexed by vectors in $\mathcal{U}$ and whose columns are indexed by vectors in $\mathcal{V}$ such that:

$$M_{i,j} = M(\boldsymbol{u}^{(i)}, \boldsymbol{v}^{(j)}) \;:=\; f(\boldsymbol{u}_1^{(i)} \boldsymbol{v}_1^{(j)}, \ldots, \boldsymbol{u}_{n'}^{(i)} \boldsymbol{v}_{n'}^{(j)}) \,.$$

Note that $M(\boldsymbol{u}^{(i)}, \boldsymbol{v}^{(j)}) \neq 0$ iff $i = j$. Hence, M is full rank diagonal matrix (infact, it is the identity matrix!).

On the other hand, by Lucas' Theorem, f has degree at most $p^s - 1 = m - 1$ and n' variables. Notice that there are at most $\binom{n'}{m-1} + \binom{n'}{m-2} + \ldots + \binom{n'}{0} \leq \binom{n'+m-1}{m-1}$ possible monomials up to degree $m-1$. For any polynomial f of degree $m-1$ we can consider the vector $\widetilde{\boldsymbol{f}} \in \mathbb{Z}_m^{\binom{n'+m-1}{m-1}}$, a vector of coefficients of each of $\binom{n'+m-1}{m-1}$ monomials. One can also take any input $x_1, \ldots, x_{n'}$ and consider a vector $\widetilde{\boldsymbol{x}} \in \mathbb{Z}_m^{\binom{n'+m-1}{m-1}}$ consisting of evaluations of each of the monomials on $x_1, \ldots, x_{n'}$. Notice that then $f(x_1, \ldots, x_{n'}) = \langle \widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{f}} \rangle$. Similarly, for each $\boldsymbol{u} \in \mathcal{U}, \boldsymbol{v} \in \mathcal{V}$ we can find $\widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{v}} \in \mathbb{Z}_m^{\binom{n'+m-1}{m-1}}$, such that $f(\boldsymbol{u}_1 \boldsymbol{v}_1, \ldots, \boldsymbol{u}_{n'} \boldsymbol{v}_{n'}) = \langle \widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{v}} \rangle$, for example $\widetilde{\boldsymbol{u}}$ consists of evaluations of monomials on input $u_1, \ldots, u_{n'}$ while $\widetilde{\boldsymbol{v}}$ combines evaluations of monomials on $v_1, \ldots, v_{n'}$ with coefficients of polynomial f. Thus, an obvious counting argument gives the desired upper bound:

$$t \;=\; \operatorname{rank}(M) \;\leq\; \binom{n'+m-1}{n'} \;=\; \binom{nm^2+m-1}{m-1} \;\leq\; (2enm)^{m-1} \,. \qquad \blacktriangleleft$$

Note that the above proof fails immediately for composites. For one, we don't have a version of the Lucas' Theorem. Moreover, there is no unified notion of rank that can be utilized, for example by reducing it to working over a prime field.

▶ Remark 22. For any constant $m = p^s$ for some prime p and positive integer s, Theorem 21 implies $\mathbf{MV}(m, n, r) \leq (2emn)^{m-1} = \operatorname{poly}(n)$ for any $0 < r \leq m$.

2.5 Some properties of 3-restricted matching vector families

We will consider matching vector families in $\mathbb{Z}_m$ where m is a composite positive integer. All algebraic operations in this section, unless otherwise stated, are modulo $\mathbb{Z}_m$.

Let $\alpha, \beta \in \mathbb{Z}_m \setminus \{0\}$. Consider a matching vector family $(\mathcal{U}, \mathcal{V})$ such that $\mathcal{U} = \{u_1, \ldots, u_K\}, V = \{v_1, \ldots, v_K\} \subseteq \mathbb{Z}_m^n$ that satisfy the following properties:

- For all $i \in [K]$, $\langle \boldsymbol{u}_i, \boldsymbol{v}_i \rangle = 0$.
- For all $i, j \in [K]$ with $i \neq j$, we have that $\langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle \in \{\alpha, \beta\}$.

We call $(\mathcal{U}, \mathcal{V})$ an $\{\alpha, \beta\}$-matching vector family of size K.

▶ Remark 23. For the rest of the paper until the end of Section 3.2, we will assume that $(\mathcal{U}, \mathcal{V})$ is not a matching vector family modulo m' for any $m'|m$. We can make this assumption without loss of generality. This is because we will prove an upper bound on K that increases as a function of m, and we get a better bound if there exists such an m'.

In our argument, it will be useful to work with composites that are exactly the product of two prime powers. This is clearly the first non-trivial case since we have a polynomial upper bound (in n) for the case of prime powers. In fact, for 3-restricted matching vector families, [7] demonstrated that this is the only non-trivial case. We include the proof below for completeness.

▶ **Lemma 24** ([7]). *Let $(\mathcal{U}, \mathcal{V})$ be a $(q+1)$-restricted matching vector family in $\mathbb{Z}_m^n$. Then m has at most q prime factors.*

Proof. Suppose $m = \prod_i p_i^{r_i}$ for some primes p_i's and corresponding positive integers r_i's. Suppose the possible non-zero values of inner products are $\alpha_1, \ldots, \alpha_q \pmod m$. For each α_j, there must exist i such that $\alpha_j \neq 0 \pmod{p_i^{r_i}}$. We can take m' to be the product of atmost q prime powers that satisfy the above condition for the respective α_j's. Thus the $(q+1)$-restricted matching vector family in $\mathbb{Z}_m^n$ induces (via the modulo m' operation) a matching vector family in $\mathbb{Z}_{m'}^n$ of the same size, and with the same number of restrictions. ◀

From the previous lemma, we can assume $m = p^s q^t$ for any $\{\alpha, \beta\}$-matching vector family $(\mathcal{U}, \mathcal{V})$ where p, q are some primes and s, t are some positive integers. In the next lemma, we can show some more restrictions on the values taken by α and β.

▶ **Lemma 25.** *Let $m = p^s q^t$ be a composite positive integer such that for any m' which is a divisor of m, $(\mathcal{U}, \mathcal{V})$ is not a matching vector family modulo m'. Then we have $\alpha = 0 \pmod{p^s q^{t-1}}$, $\alpha \neq 0 \pmod{q^t}$, $\beta = 0 \pmod{p^{s-1} q^t}$, $\beta \neq 0 \pmod{p^s}$.*

Proof. By the same argument as Lemma 24, we can assume without loss of generality that $\alpha \neq 0 \pmod{q^t}$ and $\beta \neq 0 \pmod{p^s}$. Now, suppose for a contradiction $\beta \neq 0 \pmod{p^{s-1} q^t}$. Clearly, $\alpha \neq 0 \pmod{p^{s-1} q^t}$ Then $(\mathcal{U}, \mathcal{V})$ is a $\{\alpha, \beta\}$-matching vector family modulo $m' = p^{s-1} q^t \mid m$ which is a contradiction. Similarly, we can show $\alpha = 0 \pmod{p^s q^{t-1}}$. ◀

3 Upper Bound for 3-restricted Matching Vector Families

Let $\mathcal{U}, \mathcal{V}$ be a matching vector family of size K in $\mathbb{Z}_m^n$ with non-zero residues α and β. By Lemma 24, we have that $m = p^s q^t$ for primes p and q and integers $s, t \geq 1$ with $p < q$. Further, using Lemma 25, we have that $\alpha = 0 \pmod{p^s q^{t-1}}$, $\alpha \neq 0 \pmod{q^t}$, $\beta = 0 \pmod{p^{s-1} q^t}$, $\beta \neq 0 \pmod{p^s}$. Define, a parameter τ (which is a small constant) as follows:

$$\tau := \frac{1}{2pq}.$$

Let $\mathcal{S}_{q,K} = \{A \subset [K] \mid |A| = q\}$ be the set of all subsets of $[K]$ of size q. Let $\omega = e^{\frac{2\pi i}{m}}$ be a primitive root of unity of order m.

For analysis, we define a few random variables as follows: Let $\boldsymbol{U}_i$ be uniform and independent vectors drawn from $\mathcal{U}$. Let ℓ be the *smallest* integer such that $q \leq 2^\ell$. We have:

1. $\boldsymbol{U} := \boldsymbol{U}_i$.
2. $\boldsymbol{U}^{\S} := \sum_{i=1}^{2^\ell} \boldsymbol{U}_i - \sum_{i=2^\ell+1}^{2^{\ell+1}} \boldsymbol{U}_i$. That is, sample *with replacement* $2 \cdot 2^\ell$ independent vectors from $\mathcal{U}$, add up the first 2^ℓ vectors and subtract away the remaining 2^ℓ vectors.

3. $\boldsymbol{U}^{\dagger} := \sum_{i=1}^{q} \boldsymbol{U}_i$. That is, sample *with replacement* q independent vectors from $\mathcal{U}$ and add them up.
4. $\boldsymbol{U}^{\ddagger} := \sum_{i=q+1}^{2^\ell} \boldsymbol{U}_i - \sum_{i=2^\ell+1}^{2^{\ell+1}} \boldsymbol{U}_i$. Here we define $\boldsymbol{U}^{\ddagger}$ such that $\boldsymbol{U}^{\S} = \boldsymbol{U}^{\dagger} + \boldsymbol{U}^{\ddagger}$.
5. $\boldsymbol{U}^{\star}$ is distributed as follows: pick a subset $A \sim \mathcal{S}_{q,K}$ uniformly at random and $\boldsymbol{U}^{\star} := \sum_{i\in A} \boldsymbol{u}_i$. In other words, sample *without replacement* q vectors from $\mathcal{U}$ and add them up. This is in contrast to $\boldsymbol{U}^{\dagger}$ which is sampling *with replacement*.

Random variables for $\mathcal{V}$ are defined similarly.

Sampling without replacement helps in lower bounding the min-entropy of $\boldsymbol{U}^{\star}$. On the other hand, we upper bound the min-entropy of $\boldsymbol{U}^{\dagger}$ where sampling with replacement (and hence full independence) is useful. And we are able to relate these two quantities as these two distributions are really close to each other.

The introduction of $\boldsymbol{U}^{\S}$ (and hence $\boldsymbol{U}^{\ddagger} = \boldsymbol{U}^{\S} - \boldsymbol{U}^{\dagger}$) serves as a tool for lower bounding the bias of $\boldsymbol{U}^{\dagger}$ via Lemma 13 because we are restricted to sums containing powers of 2 elements.

3.1 Upper Bound on Min-Entropy

In the following lemma, we prove that since $\mathcal{U}, \mathcal{V}$ constitute a 3-restriced **MV** family over $\mathbb{Z}_m^n$, then $\langle \boldsymbol{U}, \boldsymbol{V}\rangle$ has to be *far* from uniform. This is captured via Fourier analysis.

▶ **Lemma 26** (Large bias for **MV** family). *Let $K \geq 10pq$. Then, we have that $\left|\mathbb{E}[\omega^{\langle \boldsymbol{U},\boldsymbol{V}\rangle}]\right| \geq \tau$.*

Proof. This proof relies on the fact that the random variable $\omega^{\langle \boldsymbol{U},\boldsymbol{V}\rangle}$ is essentially supported on two values, $\{\omega^\alpha, \omega^\beta\}$, since the mass on ω^0 (corresponding to $\langle \boldsymbol{U}, \boldsymbol{V}\rangle = 0$) is very small. To complete the argument, we show that $\alpha - \beta$ is an integer different from $m/2$, and thus the mass on ω^α doesn't cancel that on ω^β. The detailed argument is given below.

Recall that $\langle \boldsymbol{U}, \boldsymbol{V}\rangle \in \{0, \alpha, \beta\}$ and $\Pr[\langle \boldsymbol{U}, \boldsymbol{V}\rangle = 0] = \frac{1}{K}$. Let $\lambda := \Pr[\langle \boldsymbol{U}, \boldsymbol{V}\rangle = \alpha]$, $\lambda \in \left[0, 1 - \frac{1}{K}\right]$. Then $\Pr[\langle \boldsymbol{U}, \boldsymbol{V}\rangle = \beta] = 1 - \lambda - \frac{1}{K}$. Since $\alpha = 0 \pmod{p^s q^{t-1}}$ and $\beta = 0 \pmod{p^{s-1}q^t}$, let $\frac{\alpha}{m} = \frac{c}{q}$ and $\frac{\beta}{m} = \frac{d}{p}$, for some $0 < c < q$ and $0 < d < p$. By definition of the bias, we have

$$
\begin{aligned}
\left|\mathbb{E}[\omega^{\langle \boldsymbol{U},\boldsymbol{V}\rangle}]\right| &= \left|\sum_{u,v\in\mathbb{Z}_m^n} \boldsymbol{U}(u)\boldsymbol{V}(v)\omega^{\langle u,v\rangle}\right| \\
&\geq \min_{\lambda\in[0,1-\frac{1}{K}]} \left|\frac{1}{K} + \lambda\omega^\alpha + (1 - 1/K - \lambda)\omega^\beta\right| \\
&\geq \min_{\lambda\in[0,1-\frac{1}{K}]} \left|\lambda\omega^\alpha + (1 - 1/K - \lambda)\omega^\beta\right| - \frac{1}{K} \\
&= \min_{\lambda\in[0,1-\frac{1}{K}]} \left|\lambda + (1 - 1/K - \lambda)\omega^{\beta-\alpha}\right| - \frac{1}{K} \\
&= \min_{\lambda\in[0,1-\frac{1}{K}]} \sqrt{\lambda^2 + (1-1/K-\lambda)^2 + 2\lambda(1-1/K-\lambda)\cos\left(2\pi\left(\frac{c}{q} - \frac{d}{p}\right)\right)} - \frac{1}{K} \\
&\geq \min_{\lambda\in[0,1-\frac{1}{K}]} \sqrt{\lambda^2 + (1-1/K-\lambda)^2 + 2\lambda(1-1/K-\lambda)\cos\left(\frac{(pq-1)\pi}{pq}\right)} - \frac{1}{K} \\
&\geq \left(\frac{K-1}{K}\right)\cdot\sin\left(\frac{\pi}{2pq}\right) - \frac{1}{K} \\
&\geq \tau\,.
\end{aligned}
$$

The third equality follows from the identity that for constants c_i, and θ,

$$|c_1 + c_2 e^{i\theta}|^2 = |(c_1 + c_2 \cos\theta) + ic_2 \sin\theta|^2 = c_1^2 + c_2^2 + 2c_1 c_2 \cos\theta \,.$$

The third last inequality follows from the observation (which we discuss below) that $\left|2\pi\frac{cp-dq}{pq} - \pi\right| \geq \frac{\pi}{pq}$, and that cos θ for $\theta \in (0, 2\pi)$ is minimized when $|\theta - \pi|$ is minimized. If pq is odd, the above is trivially true since after multiplying left-hand-side by $\frac{pq}{\pi}$ we obtain $|2(cp - dq) - pq|$ and this is a difference of odd and even number and thus the absolute value has to be at least 1. Now, suppose $p = 2$, via similar argument we have

$$\left|\frac{2c - dq}{q} - 1\right| \geq \frac{1}{q} > \frac{1}{2q} \,,$$

since $c \neq 0 \pmod q$ and q is an odd prime.

The second last inequality follows from taking $\lambda = \frac{K-1}{2K}$ which minimizes the expression, and using $1 - \cos\theta = 2\sin^2(\theta/2)$, giving a value of the square root to be

$$\left(2\left(\frac{K-1}{2K}\right)^2 - 2\cdot\left(\frac{K-1}{2K}\right)^2 \cos\left(\frac{\pi}{pq}\right)\right)^{1/2} = \left(\frac{K-1}{K}\right)\sin\left(\frac{\pi}{2pq}\right) \,.$$

Finally, the last inequality follows from $\sin x \geq \frac{x}{2}$ for $x \in [0, 1]$ and that $K > 10pq$. ◀

Next, we will show that $\boldsymbol{U}^\star$ has low min-entropy.

▶ **Lemma 27.** $\left|\mathbb{E}[\omega^{\langle \boldsymbol{U}^\dagger + \boldsymbol{U}^\ddagger, \boldsymbol{V}^\dagger + \boldsymbol{V}^\ddagger\rangle}]\right| \geq \tau^{16q^2}$.

Proof. Recall that ℓ is the *smallest* integer such that $q \leq 2^\ell$. Thus, $2^{\ell-1} < q$, i.e., $2^\ell < 2q$. By Lemma 13, we can control the bias of $\langle \boldsymbol{U}^\S, \boldsymbol{V}^\S\rangle$:

$$\begin{aligned}
\left|\mathbb{E}[\omega^{\langle \boldsymbol{U}^\dagger + \boldsymbol{U}^\ddagger, \boldsymbol{V}^\dagger + \boldsymbol{V}^\ddagger\rangle}]\right| &= \left|\mathbb{E}[\omega^{\langle \boldsymbol{U}^\S, \boldsymbol{V}^\S\rangle}]\right| \\
&\geq \left|\mathbb{E}[\omega^{\langle \boldsymbol{U}, \boldsymbol{V}\rangle}]\right|^{2^{2(\ell+1)}} \\
&\geq \left|\mathbb{E}[\omega^{\langle \boldsymbol{U}, \boldsymbol{V}\rangle}]\right|^{4\cdot(2q)^2} \\
&= \tau^{16q^2} \,.
\end{aligned}$$

◀

Define the event E_U (resp. E_V) where each $\boldsymbol{U}_i$ (resp. $\boldsymbol{V}_i$) is unique when $\boldsymbol{U}^\dagger$ (resp. $\boldsymbol{V}^\dagger$) is sampled. Notice that $\boldsymbol{U}^\dagger | E_U$ is distributed exactly as $\boldsymbol{U}^\star$. Also, $\Pr[\neg E_U] = \Pr[\neg E_V] \leq \frac{q^2}{K}$ and so, by the union bound $\Pr[\neg(E_U \wedge E_V)] \leq \frac{2q^2}{K}$.

▶ **Lemma 28.** $\left|\mathbb{E}[\omega^{\langle \boldsymbol{U}^\star + \boldsymbol{U}^\ddagger, \boldsymbol{V}^\star + \boldsymbol{V}^\ddagger\rangle}]\right| \geq \tau^{16q^2} - \frac{2q^2}{K}$.

Proof. The proof uses the above observation that $\boldsymbol{U}^\dagger | E_U$ (resp. $\boldsymbol{V}^\dagger | E_V$) is distributed exactly as $\boldsymbol{U}^\star$ (resp. $\boldsymbol{V}^\star$), and that both E_U and E_V hold simultaneously with probability at least $1 - \frac{2q^2}{K}$. In particular,

$$\begin{aligned}
\left|\mathbb{E}[\omega^{\langle \boldsymbol{U}^\dagger+\boldsymbol{U}^\ddagger,\boldsymbol{V}^\dagger+\boldsymbol{V}^\ddagger\rangle}]\right| &= \Big|\mathbb{E}[\omega^{\langle \boldsymbol{U}^\dagger+\boldsymbol{U}^\ddagger,\boldsymbol{V}^\dagger+\boldsymbol{V}^\ddagger\rangle} \mid E_U \wedge E_V] \cdot \Pr[E_U \wedge E_V] \\
&\quad + \mathbb{E}[\omega^{\langle \boldsymbol{U}^\dagger+\boldsymbol{U}^\ddagger,\boldsymbol{V}^\dagger+\boldsymbol{V}^\ddagger\rangle} \mid \neg(E_U \wedge E_V)] \Pr[\neg(E_U \wedge E_V)]\Big| \\
&\leq \left|\mathbb{E}[\omega^{\langle \boldsymbol{U}^\dagger+\boldsymbol{U}^\ddagger,\boldsymbol{V}^\dagger+\boldsymbol{V}^\ddagger\rangle} \mid E_U \wedge E_V]\right| \Pr[E_U \wedge E_V] \\
&\quad + \left|\mathbb{E}[\omega^{\langle \boldsymbol{U}^\dagger+\boldsymbol{U}^\ddagger,\boldsymbol{V}^\dagger+\boldsymbol{V}^\ddagger\rangle} \mid \neg(E_U \wedge E_V)]\right| \Pr[\neg(E_U \wedge E_V)] \\
&= \left|\mathbb{E}[\omega^{\langle \boldsymbol{U}^\star+\boldsymbol{U}^\ddagger,\boldsymbol{V}^\star+\boldsymbol{V}^\ddagger\rangle}]\right| \Pr[E_U \wedge E_V] \\
&\quad + \left|\mathbb{E}[\omega^{\langle \boldsymbol{U}^\dagger+\boldsymbol{U}^\ddagger,\boldsymbol{V}^\dagger+\boldsymbol{V}^\ddagger\rangle} \mid \neg(E_U \wedge E_V)]\right| \Pr[\neg(E_U \wedge E_V)] \\
&\leq \left|\mathbb{E}[\omega^{\langle \boldsymbol{U}^\star+\boldsymbol{U}^\ddagger,\boldsymbol{V}^\star+\boldsymbol{V}^\ddagger\rangle}]\right| + \frac{2q^2}{K} \,.
\end{aligned}$$

It follows that $\left|\mathbb{E}[\omega^{\langle \boldsymbol{U}^\star+\boldsymbol{U}^\ddagger,\boldsymbol{V}^\star+\boldsymbol{V}^\ddagger\rangle}]\right| \geq \tau^{16q^2} - \frac{2q^2}{K}$. ◀

▶ **Lemma 29.** $\mathbf{H}_\infty(\boldsymbol{U}^\star) + \mathbf{H}_\infty(\boldsymbol{V}^\star) \leq n \log m - 2\log(\tau^{16q^2} - \frac{2q^2}{K})$.

Proof. By Lemma 28 and Lemma 10, we have that

$$\mathbf{H}_2(\boldsymbol{U}^\star + \boldsymbol{U}^\ddagger) + \mathbf{H}_2(\boldsymbol{V}^\star + \boldsymbol{V}^\ddagger) \leq n \log m - 2\log(\tau^{16q^2} - \frac{2q^2}{K}) \,.$$

Finally, notice that $\boldsymbol{U}^\star$ is independent of $\boldsymbol{U}^\ddagger$ and $\boldsymbol{V}^\star$ is independent of $\boldsymbol{V}^\ddagger$. By Lemma 9, we have:

$$\mathbf{H}_\infty(\boldsymbol{U}^\star) + \mathbf{H}_\infty(\boldsymbol{V}^\star) \leq \mathbf{H}_\infty(\boldsymbol{U}^\star + \boldsymbol{U}^\ddagger) + \mathbf{H}_\infty(\boldsymbol{V}^\star + \boldsymbol{V}^\ddagger) \leq n \log m - 2\log\left(\tau^{16q^2} - \frac{2q^2}{K}\right) \,,$$

where $\mathbf{H}_\infty(X) \leq \mathbf{H}_2(X)$ for any random variable X. ◀

3.2 Lower Bound on Min-Entropy

In this subsection, we will talk about sums of vectors from $\mathcal{U}, \mathcal{V}$. For each $A \subseteq [K]$, we can naturally associate a sum of vectors $\sum_{i \in A} \boldsymbol{u}_i$ from $\mathcal{U}$ (and similarly for $\mathcal{V}$). Note that we only care about *unique* sums where each vector from $\mathcal{U}$ (or $\mathcal{V}$) appears only once. In this view, a *collision* corresponds to pair of sets $A, B, A \neq B$ such that $\sum_{i \in A} \boldsymbol{u}_i = \sum_{j \in B} \boldsymbol{u}_j$ (or similarly for vectors from $\mathcal{V}$). In the following we give a lower bound on the min-entropy of $\boldsymbol{U}^\star$. By symmetry, the statement holds for $\boldsymbol{V}^\star$ as well.

We start with a lemma that discusses the behaviour of a collision.

▶ **Lemma 30.** *Let $m = p^s q^t$ where $q > p$ and $\mathcal{U}, \mathcal{V}$ be a $\{0, \alpha, \beta\}$-restricted matching vector family of size K such that $\alpha = 0 \pmod{p^s q^{t-1}}$, $\alpha \neq 0 \pmod{q^t}$ and $\beta = 0 \pmod{p^{s-1} q^t}$, $\beta \neq 0 \pmod{p^s}$. Let $A, B \in \mathcal{S}_{q,K}, A \neq B$ be such that $\sum_{i \in A} \boldsymbol{u}_i = \sum_{i \in B} \boldsymbol{u}_i$ where each $\boldsymbol{u}_i \in \mathcal{U}$. Then:*

- **(i)** *For each $i \in A, j \in B$, $\boldsymbol{u}_i \neq \boldsymbol{u}_j$ $(A \cap B = \emptyset)$.*
- **(ii)** *For each $i \in A, j \in B$, $\langle \boldsymbol{u}_j, \boldsymbol{v}_i \rangle = \alpha \pmod{q^t}$.*

Proof.

Proof of Part (i). For any $i \in A$, we can write $\sum_{j\in B}\langle \boldsymbol{u}_j, \boldsymbol{v}_i\rangle - \sum_{j\in A}\langle \boldsymbol{u}_j, \boldsymbol{v}_i\rangle$ as a linear combination of α's and β's such that

$$a_i\alpha + b_i\beta = \sum_{j\in B}\langle \boldsymbol{u}_j, \boldsymbol{v}_i\rangle - \sum_{j\in A}\langle \boldsymbol{u}_j, \boldsymbol{v}_i\rangle = \langle \sum_{j\in B}\boldsymbol{u}_j - \sum_{j\in A}\boldsymbol{u}_j, \boldsymbol{v}_i\rangle = 0 \pmod m \tag{1}$$

for some integers a_i, b_i.

As $\langle \boldsymbol{u}_i, \boldsymbol{v}_i\rangle = 0 \pmod m$, there are q terms in the first sum and $q-1$ terms in the second sum; this is where we use the fact that $\boldsymbol{u}_i$ were picked without replacement. Therefore, we have the obvious inequalities: $-(q-1) \le a_i, b_i \le q$. Since, $\beta = 0 \pmod{q^t}$, and $a_i\alpha + b_i\beta = 0 \pmod m$, it follows that $a_i\alpha + b_i\beta = 0 \pmod{q^t} \implies a_i\alpha = 0 \pmod{q^t}$. Since α is divisible by q^{t-1}, and not q^t, and a_i is between $-(q-1)$ and q, it must happen that either $a_i = 0$ or $a_i = q$.

Suppose for a contradiction that $i \in A \cap B$. Let $k \in A\backslash B$ (non-empty since $A \neq B$). Then,

$$a_k\alpha + b_k\beta = \sum_{j\in B}\langle \boldsymbol{u}_j, \boldsymbol{v}_k\rangle - \sum_{j\in A}\langle \boldsymbol{u}_j, \boldsymbol{v}_k\rangle = \sum_{j\in B\backslash A}\langle \boldsymbol{u}_j, \boldsymbol{v}_k\rangle - \sum_{j\in A\backslash(B\cup\{k\})}\langle \boldsymbol{u}_j, \boldsymbol{v}_k\rangle .$$

By our choice of $\boldsymbol{u}_k$, fact that all inner products on the right-hand side above are non-zero, and since $|A| = |B| = q$, we know that:

1. $|B\backslash A| = |A\backslash(B\cup\{k\})| + 1$, and therefore $a_k + b_k = |B\backslash A| - |A\backslash(B\cup\{k\})| = 1$.
2. $-(q-1) \le -|A\backslash(B\cup\{k\})| \le b_k \le |B\backslash A| \le q-1$

If $a_k = 0$, then $b_k = 1$ implies that $a_k\alpha + b_k\beta = \beta \neq 0 \pmod m$, a contradiction to Equation (1). Also, notice that $a_k \neq q$ since for the same reason as above $a_k \le |B\backslash A| \le q-1$. This completes the proof of Part (i).

Proof of Part (ii). Now, consider any $i \in A$, clearly $i \notin B$ from our proof above. From a similar argument as above, we know $a_i + b_i = 1$. And further $a_i = q$ and $b_i = 1 - a_i = -(q-1)$. However the only way we can have $a_i = q$ is if each term in the sum $\sum_{j\in B}\langle \boldsymbol{u}_j, \boldsymbol{v}_i\rangle$ is α which completes the proof. Note that we can trivially reduce from mod m to mod q^t, since $\langle \boldsymbol{u}_j, \boldsymbol{v}_i\rangle = \alpha \neq 0 \pmod{q^t}$. ◀

▶ **Theorem 31.** *Let $m = p^s q^t$ where $q > p$ and let $\boldsymbol{U}^\star$ be defined as in the beginning of this section. Then $\mathbf{H}_\infty(\boldsymbol{U}^\star) \ge q\log\left(\frac{K}{q}\right) - \log(nt+1)$.*

Proof. We will upper bound the probability $\Pr_{\boldsymbol{U}^\star}[\boldsymbol{U}^\star = \boldsymbol{u}]$, where $\boldsymbol{u} \in \mathbb{Z}_m^n$ is an arbitrary vector. Let

$$C_{\boldsymbol{u}} := \left\{ A \in \mathcal{S}_{q,K} \mid \sum_{i\in A}\boldsymbol{u}_i = \boldsymbol{u} \right\},$$

be the set of sums of length q from $\mathcal{S}_{q,K}$ such that its sum is equal to $\boldsymbol{u}$. Note that each term in the sum is unique by how we sample $\boldsymbol{U}^\star$.

Now, consider any pair of collisions $A, B \in C_{\boldsymbol{u}}$. From Lemma 30, we know that for each $i \in A, j \in B$, $i \neq j$ and $\langle \boldsymbol{u}_j, \boldsymbol{v}_i\rangle = \alpha$, and all the sets in $C_{\boldsymbol{u}}$ are disjoint. In particular, if we pick exactly one vector from each sum in $C_{\boldsymbol{u}}$ (and their corresponding matching vectors in $\mathcal{V}$), the set of vectors form a $\{\alpha\}$-matching vector family in $\mathbb{Z}_{q^t}^n$. Formally, $\mathcal{U}' := \{\boldsymbol{u}_i \mid i = \min(A), A \in C_{\boldsymbol{u}}\} \subseteq \mathcal{U}$, and $\mathcal{V}' \subseteq \mathcal{V}$ with the corresponding matching vectors forms a $\{\alpha\}$-matching vector family in $\mathbb{Z}_{q^t}^n$ of size $|C_{\boldsymbol{u}}|$.

By Lemma 18, we have $|C_{\boldsymbol{u}}| \leq nt + 1$. Therefore, we have

$$\mathbf{H}_\infty(\boldsymbol{U}^\star) \;\geq\; \log\left(\frac{\binom{K}{q}}{nt+1}\right) \;\geq\; q\log\left(\frac{K}{q}\right) - \log(nt+1)\;. \qquad \blacktriangleleft$$

▶ Remark 32. As pointed out by a helpful anonymous reviewer, we can extend Lemma 30 to work for $A, B \in \mathcal{S}_{q',K}$ where $q' > q$ in certain cases. Suppose we again write $\sum_{j\in B}\langle \boldsymbol{u}_j, \boldsymbol{v}_i\rangle - \sum_{j\in A}\langle \boldsymbol{u}_j, \boldsymbol{v}_i\rangle$ as a linear combination of $a_{i,q'}\alpha + b_{i,q'}\beta = 0 \pmod m$. The proof only requires two properties from q':

1. For the first part, we require that for every $q'' < q'$ the corresponding equation $a_{i,q''}\alpha + b_{i,q''}\beta = 0 \pmod m$ has only trivial solutions ($a_{i,q''} = b_{i,q''} = 0$).
2. For the second part, we require that $a_{i,q'} = q'$.

For $m = 6$, $q' = 3$ but by a calculation it can be shown that for $m = 15$, we can have $q' = 6 > 5 = q$. This implies that we can improve the conclusion of Theorem 31 to $\mathbf{H}_\infty(\boldsymbol{U}^\star) \geq q'\log\left(\frac{K}{q'}\right) - \log(nt+1)$ which eventually gives us a better bound for Theorem 33. However, we omit this for clarity.

3.3 Main Theorem

We are now ready to prove our main theorem.

▶ **Theorem 33.** *Let $n, m \geq 1$ be positive integers with $n > 10m^3$. Suppose $\mathcal{U}, \mathcal{V}$ is a 3-restricted* **MV** *family of size K over $\mathbb{Z}_m^n$. Then $K \leq (2emn)^{m-1}$ if m is a prime power, and otherwise*

$$K \;\leq\; m^{\frac{n+2\log(n+1)}{2q}+2m^2}\;,$$

where q is the second-smallest prime dividing m.

Proof. Let m' be the smallest factor of m such that $\mathcal{U}, \mathcal{V}$ is a 3-restricted **MV** family of size K over $\mathbb{Z}_{m'}^n$. If m' is a prime power, then by Theorem 21,

$$K \;\leq\; (2em'n)^{m'-1} \;\leq\; (2emn)^{m-1}\;.$$

Suppose m is not a prime power, then $K \leq (2emn)^{m-1} \leq m^{\frac{n+2\log(n+1)}{2q}+2m^2}$; the second inequality follows, since $n > 10m^3$.

Now we consider the case when m' is not a prime power. Then, by Lemma 24, we have that $m' = p_1^s \cdot p_2^t$ for primes p_1 and p_2 and integers $s, t \geq 1$ with $p_1 < p_2$. Further, using Lemma 25, we have that $\alpha = 0 \pmod{p_1^s p_2^{t-1}}$, $\alpha \neq 0 \pmod{p_2^t}$, $\beta = 0 \pmod{p_1^{s-1}p_2^t}$, $\beta \neq 0 \pmod{p_1^s}$.

Let q be the *second-smallest* prime dividing m, and we assume that $K > m^{\frac{n+2\log(n+1)}{2q}+2m^2}$, since otherwise we are already done. Also, note that by assumption, $p_2 \geq q$. Since $n \geq 10m^3$, and $2 \leq p_1p_2 \leq m$, this implies that

$$K \;\geq\; m^{7m^2} \;\geq\; m^2\cdot(2m)^{4m^2} \;\geq\; 4p_2^2\cdot(2p_1p_2)^{16p_2^2}\;.$$

By Lemma 29, we have without loss of generality (between $\boldsymbol{U}^\star$ and $\boldsymbol{V}^\star$) that

$$\begin{aligned}
\mathbf{H}_\infty(\boldsymbol{U}^\star) &\leq \frac{n\log m'}{2} - \log\left(\left(\frac{1}{2p_1p_2}\right)^{16p_2^2} - \frac{2p_2^2}{K}\right)\\
&\leq \frac{n\log m}{2} - \log\left(\frac{1}{2}\cdot\left(\frac{1}{2p_1p_2}\right)^{16p_2^2}\right)\\
&\leq \frac{n\log m}{2} + 16p_2^2\cdot\log(2p_1p_2) + 1\\
&\leq \frac{n\log m}{2} + 4m^2\log m + 4m^2 + 1\,.
\end{aligned}$$

Here we used that $2p_2 \leq p_1p_2 \leq m$. On the other hand, by Theorem 31, we know that

$$\mathbf{H}_\infty(\boldsymbol{U}^\star) \geq p_2\log\left(\frac{K}{p_2}\right) - \log(nt+1)\,.$$

Putting them together we have:

$$p_2\log\left(\frac{K}{p_2}\right) - \log(nt+1) \leq \frac{n\log m}{2} + 4m^2\log m + 4m^2 + 1\,.$$

This gives us that

$$\begin{aligned}
K &\leq p_2\cdot m^{\frac{n}{2p_2}}\cdot m^{\frac{4m^2}{p_2}}\cdot 2^{\frac{4m^2+1}{p_2}}\cdot 2^{\frac{\log(nt+1)}{p_2}}\\
&\leq m^{\frac{n}{2p_2}}\cdot m^{\frac{4m^2}{p_2}+1}\cdot m^{\frac{1.9m^2}{p_2}}\cdot 2^{\frac{\log(nt+1)}{p_2}}\\
&\leq m^{\frac{n}{2p_2}}\cdot m^{\frac{6m^2}{p_2}}\cdot 2^{\frac{\log(nt+1)}{p_2}}\\
&\leq m^{\frac{n}{2p_2}}\cdot m^{2m^2}\cdot 2^{\frac{\log(nt+1)}{p_2}}\\
&\leq m^{\frac{n}{2q}}\cdot m^{2m^2}\cdot 2^{\frac{\log(nt+1)}{p_2}}\\
&= m^{\frac{n}{2q}}\cdot m^{2m^2}\cdot m^{\frac{\log(nt+1)}{p_2\cdot\log m}};\\
&\leq m^{\frac{n}{2q}}\cdot m^{2m^2}\cdot m^{\frac{\log(nt+1)}{p_2\cdot t\cdot\log p_2}};\\
&\leq m^{\frac{n}{2q}}\cdot m^{2m^2}\cdot m^{\frac{\log(nt+1)}{p_2\cdot t}}\\
&\leq m^{\frac{n}{2q}}\cdot m^{2m^2}\cdot m^{\frac{\log(nt+1)^{1/t}}{p_2}}\\
&\leq m^{\frac{n}{2q}}\cdot m^{2m^2}\cdot m^{\frac{\log(n+1)}{p_2}}\\
&\leq m^{\frac{n+2\log(n+1)}{2q}}\cdot m^{2m^2}\,.
\end{aligned}$$

In the second inequality, we used $p_2 \leq m$, and also, $2^{4m^2+1} < m^{1.9m^2}$, which is true since $m \geq 6$. In the third inequality, we used the fact that $\frac{6m^2}{p_2} \geq \frac{5.9m^2}{p_2} + 1$, which is equivalent to showing $\frac{m^2}{10p_2} \geq 1$, which is true since $\frac{m^2}{10p_2} = \left(\frac{m}{2p_2}\right)\cdot\left(\frac{m}{5}\right) > 1$, because $m \geq 2p_2 \geq 2q \geq 6$. In the seventh inequality, we used that $m \geq p_2^t$. Finally, the second to last inequality follows from Bernoulli's inequality: $(nt+1) \leq (n+1)^t$.

This is a contradiction to our assumption that $K > m^{\frac{n+2\log(n+1)}{2q}+2m^2}$. This finishes the proof. ◀

▶ **Corollary 34.** *For an arbitrary positive integer m, consider a 3-query matching vector codes $C : \Sigma^K \to \Sigma^N$ constructed from 3-restricted matching vector families using the construction in Theorem 15, then $N \geq \Omega(K^{6-o_K(1)})$.*

The proof follows straightforwardly from the Theorem 15 combined with Theorem 33.

References

1 Noga Alon. The shannon capacity of a union. *Combinatorica*, 18(3):301–310, 1998. doi:10.1007/PL00009824.

2 Omar Alrabiah, Venkatesan Guruswami, Pravesh K Kothari, and Peter Manohar. A near-cubic lower bound for 3-query locally decodable codes from semirandom csp refutation. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1438–1448, 2023. doi:10.1145/3564246.3585143.

3 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45(3):501–555, 1998. doi:10.1145/278298.278306.

4 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of np. *Journal of the ACM (JACM)*, 45(1):70–122, 1998. doi:10.1145/273865.273901.

5 László Babai and Péter Frankl. Linear algebra methods in combinatorics. *to appear*, 2020.

6 Arnab Bhattacharyya, L. Sunil Chandran, and Suprovat Ghoshal. Combinatorial Lower Bounds for 3-Query LDCs. In 11^{th} *Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151, pages 85:1–85:8, 2020. doi:10.4230/LIPIcs.ITCS.2020.85.

7 Abhishek Bhowmick, Zeev Dvir, and Shachar Lovett. New bounds for matching vector families. *SIAM Journal on Computing*, 43(5):1654–1683, 2014. doi:10.1137/130932296.

8 Yeow Meng Chee, San Ling, Huaxiong Wang, and Liang Feng Zhang. Upper bounds on matching families in $\mathbb{Z}_{pq}^n$. *IEEE transactions on information theory*, 59(8):5131–5139, 2013.

9 Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. *SIAM Journal on Computing*, 40(4):1154–1178, 2011. doi:10.1137/100804322.

10 Klim Efremenko. 3-query locally decodable codes of subexponential length. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 39–44, 2009. doi:10.1145/1536414.1536422.

11 P. Frankl and R. M. Wilson. Intersection theorems with geometric consequences. *Combinatorica*, 1(4):357–368, December 1981. doi:10.1007/bf02579457.

12 Oded Goldreich, Howard Karloff, Leonard J Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. *computational complexity*, 15:263–296, 2006. doi:10.1007/S00037-006-0216-3.

13 P. Gopalan. Constructing ramsey graphs from boolean function representations. In *21st Annual IEEE Conference on Computational Complexity (CCC'06)*, pages 14 pp.–128, 2006. doi:10.1109/CCC.2006.14.

14 Sivakanth Gopi. Lecture notes: Polynomial representing or mod m. https://homes.cs.washington.edu/~anuprao/pubs/codingtheory/lecture12.pdf, 2019.

15 Sivakanth Gopi. Upper bound for matching vector families for prime-powers. *Personal Communication*, 2024.

16 W. T. Gowers, Ben Green, Freddie Manners, and Terence Tao. Marton's conjecture in abelian groups with bounded torsion, 2024. arXiv:2404.02244.

17 Vince Grolmusz. On the weak mod m representation of boolean functions. *Chicago Journal of Theoretical Computer Science*, 1995.

18 Toshiya Itoh and Yasuhiro Suzuki. Improved constructions for query-efficient locally decodable codes of subexponential length. *IEICE Transactions on Information and Systems*, 93(2):263–270, 2010. doi:10.1587/TRANSINF.E93.D.263.

19 Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 80–86, 2000. doi:10.1145/335305.335315.

20 Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *Journal of Computer and System Sciences*, 3(69):395–420, 2004. doi:10.1016/J.JCSS.2004.04.007.

21 Anup Rao. An exposition of bourgain's 2-source extractor. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 14. Citeseer, 2007.

22 David P Woodruff. New lower bounds for general locally decodable codes. In *Electronic Colloquium on Computational Complexity (ECCC)*, 2007.

23 David P Woodruff. A quadratic lower bound for three-query linear locally decodable codes over any field. *Journal of Computer Science and Technology*, 27(4):678–686, 2012. doi:10.1007/S11390-012-1254-8.

24 Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *Journal of the ACM (JACM)*, 55(1):1–16, 2008. doi:10.1145/1326554.1326555.

25 Sergey Yekhanin. Locally decodable codes. *Foundations and Trends® in Theoretical Computer Science*, 6(3):139–255, 2012. doi:10.1561/0400000030.

A Proof of Lemma 11

We include the proof of completeness.

Proof.

$$|\mathbb{E}[\omega^{\langle X,Y\rangle}]| = \left|\sum_{y\in\mathbb{Z}_m^n} Y(y) \sum_{x\in\mathbb{Z}_m^n} X(x)\omega^{\langle x,y\rangle}\right|$$
$$\leq \sum_{y\in\mathbb{Z}_m^n} Y(y)\left|\sum_{x\in\mathbb{Z}_m^n} X(x)\omega^{\langle x,y\rangle}\right| .$$

On the other hand, since the square function is convex, by Jensen's inequality, one have

$$|\mathbb{E}[\omega^{\langle X,Y\rangle}]|^2 \leq \sum_{y\in\mathbb{Z}_m^n} Y(y)\left|\sum_{x\in\mathbb{Z}_m^n} X(x)\omega^{\langle x,y\rangle}\right|^2$$
$$= \left|\sum_{y\in\mathbb{Z}_m^n} Y(y) \sum_{x_1,x_2\in\mathbb{Z}_m^n} X(x_1)X(x_2)\omega^{\langle x_1,y\rangle}\omega^{-\langle x_2,y\rangle}\right|$$
$$= \left|\sum_{y\in\mathbb{Z}_m^n} Y(y) \sum_{x_1,x_2\in\mathbb{Z}_m^n} X(x_1)X(x_2)\omega^{\langle x_1-x_2,y\rangle}\right|$$
$$= \left|\mathbb{E}[\omega^{\langle X_1-X_2,Y\rangle}]\right| .$$

◀

Sparsity Lower Bounds for Probabilistic Polynomials

Josh Alman
Columbia University, New York, NY, USA

Arkadev Chattopadhyay
TIFR Mumbai, India

Ryan Williams
CSAIL and EECS, MIT, Cambridge, MA, USA

Abstract

Probabilistic polynomials over commutative rings offer a powerful way of representing Boolean functions. Although many degree lower bounds for such representations have been proved, sparsity lower bounds (counting the number of monomials in the polynomials) have not been so common. Sparsity upper bounds are of great interest for potential algorithmic applications, since sparse probabilistic polynomials are the key technical tool behind the best known algorithms for many core problems, including dense All-Pairs Shortest Paths, and the existence of sparser polynomials would lead to breakthrough algorithms for these problems.

In this paper, we prove several strong lower bounds on the sparsity of probabilistic and approximate polynomials computing Boolean functions when 0 means "false". Our main result is that the AND of n ORs of $c \log n$ variables requires probabilistic polynomials (over any commutative ring which isn't too large) of sparsity $n^{\Omega(\log c)}$ to achieve even $1/4$ error. The lower bound is tight, and it rules out a large class of polynomial-method approaches for refuting the APSP and SETH conjectures via matrix multiplication. Our other results include:

- Every probabilistic polynomial (over a commutative ring) for the disjointness function on two n-bit vectors requires exponential sparsity in order to achieve exponentially low error.
- A generic lower bound that any function requiring probabilistic polynomials of degree d must require probabilistic polynomials of sparsity $\Omega(2^d)$.
- Building on earlier work, we consider the probabilistic rank of Boolean functions which generalizes the notion of sparsity for probabilistic polynomials, and prove separations of probabilistic rank and probabilistic sparsity.

Some of our results and lemmas are basis independent. For example, over any basis $\{a, b\}$ for true and false where $a \neq b$, and any commutative ring R, the AND function on n variables has no probabilistic R-polynomial with $2^{o(n)}$ sparsity, $o(n)$ degree, and $1/2^{o(n)}$ error simultaneously. This AND lower bound is our main technical lemma used in the above lower bounds.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory; Theory of computation → Models of computation; Mathematics of computing → Discrete mathematics

Keywords and phrases Probabilistic Polynomials, Sparsity, Orthogonal Vectors, Probabilistic Rank

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.3

Funding *Josh Alman*: Work supported in part by NSF Grant CCF-2238221.
Arkadev Chattopadhyay: Supported by funds of Department of Atomic Energy, Govt. of India, under project RTI4001, and a Google India Research Award.
Ryan Williams: Work supported in part by NSF CCF-2127597 and NSF CCF-2420092.



© Josh Alman, Arkadev Chattopadhyay, and Ryan Williams;
licensed under Creative Commons License CC-BY 4.0



1 Introduction

Let R be a ring. The sparsity of a polynomial $p \in R[x_1, \ldots, x_n]$, denoted by $\text{sparsity}(p)$, is the number of monomials of p. A *probabilistic polynomial in n variables* is a distribution $\mathcal{P}$ on n-variate polynomials. Its *degree* and *sparsity* are the maximum degree and sparsity, respectively, of any polynomial in its support. The error of $\mathcal{P}$ on a function $f : \{0,1\}^n \to \{0,1\}$ is the function from $\mathbb{N}$ to $[0,1]$ defined as

$$\text{error}(\mathcal{P}, f)(n) = \max_{x \in \{0,1\}^n} \Pr_{p \sim \mathcal{P}} [p(x) \neq f(x)] .$$

Probabilistic polynomials with low sparsity are the main technical component in a number of recent randomized algorithms for core problems. This includes the best known randomized algorithm for the dense case of All-Pairs Shortest Paths (APSP) [36], the best known randomized algorithm for Orthogonal Vectors (OV) [1], the best known deterministic algorithms for these problems (which *derandomizes* the aforementioned probabilistic polynomials) [11], as well as Constraint Satisfaction problems [37, 15], All-Pairs Nearest Neighbor problems [3, 2], systems of polynomial equations over finite fields [23], online matrix-vector multiplication [22], and Stable Matching [26]. APSP and OV are especially of interest, as substantially faster algorithms for them would have many applications (see the survey of Vassilevska-Williams [38]). Moreover, it has become popular to conjecture that it is impossible to improve the best-known runtimes of these problems by polynomial factors.

For all of these algorithms, the running time improvement is completely determined by the sparsity of a certain probabilistic polynomial: if we can find a sparser polynomial, we would directly improve the runtime. This leads to the main motivating question of this paper:

Is it possible to design sparser probabilistic polynomials to speed up these algorithms further, and in particular, can we design polynomials sparse enough to refute some of the popular conjectures?

In this paper, we prove unconditionally that several known probabilistic polynomial constructions are already optimal, in a broad sense. The particular Boolean functions of interest for APSP and OV are as follows.

For a function $c : \mathbb{N} \to \mathbb{N}$, the Boolean function $OV_c : \{0,1\}^{2 \cdot n \cdot c(n) \log n} \to \{0,1\}$, is defined by:

$$OV_c(x, y) = \bigvee_{i_1, i_2 \in [n]^2} \bigwedge_{j=1}^{c(n) \log n} (\overline{x_{i_1,j}} \vee \overline{y_{i_2,j}}).$$

The best known algorithm for OV [1] uses a probabilistic $\mathbb{F}_2$-polynomial representation of the OV_c function with $n^{O(\log c)}$ sparsity to quickly check the orthogonality of many pairs of vectors. The OV problem for n input vectors of dimension $c \log n$ could be solved in truly sub-quadratic time, if a probabilistic polynomial with sparsity $n^{O(1)}$ could be designed for OV_c. In particular, if there were a polynomial for OV_c with sparsity n^k (for some universal k) that worked for every $c \geq 1$, that would yield a fast enough algorithm for OV to refute the *Strong Exponential Time Hypothesis* (SETH) [18, 10]. For completeness, we prove this in Theorem 27 in Appendix B.

Chan and Williams [36, 11] showed that $OV_{n/\log n}$ can be used to solve APSP as well. Their algorithm, the best known randomized algorithm for APSP in dense graphs, crucially uses a probabilistic $\mathbb{F}_2$-polynomial representation of the $OV_{n/\log n}$ function with $n^{O(\log n)}$ sparsity, in order to get a $n^3/2^{\Omega(\sqrt{\log n})}$ time algorithm. If the sparsity could be improved to $n^{O(1)}$, then APSP would be solvable in truly subcubic time – an algorithmic breakthrough.

Our main result shows that in a broad sense, the sparsities of the known probabilistic polynomials for OV_c are already **optimal**, up to a constant factor in the exponent:

▶ **Theorem 1.** *For every $c : \mathbb{N} \to \mathbb{N}$, every $\varepsilon > 0$ and every finite commutative ring R such that $|R| \leq 2^{O(n^{1/2-\varepsilon})}$, R-probabilistic polynomials with error $1/4$ for OV_c which use the 0 of R to represent "false" require sparsity $n^{\Omega(\log c(n))}$.*

Theorem 1 is quite powerful, as it holds for all *finite commutative rings* rather than just finite fields. For instance, many simple functions like OR are known to have smaller polynomial representations when working modulo a composite number m rather than over a finite field [6], but we rule out a sparsity improvement over such rings as well.

Theorem 1 also rules out sparser representations over larger rings (even infinite rings like $\mathbb{R}$) without very large coefficients. For instance, given a sparse probabilistic polynomial for OV_c over the integers, we could compute all of its coefficients modulo 2 to get a probabilistic polynomial over $\mathbb{F}_2$ with at most the same degree, sparsity, and error. Even if we are working over a commutative ring R where such a trick doesn't work, it takes $n^{1/2-o(1)}$ bits to describe elements of R when $|R| \geq 2^{O(n^{1/2-\varepsilon})}$. Working with such large coefficients is prohibitive when trying to solve OV or APSP, since it would presumably multiply the runtime by such a large polynomial factor, unless polynomial evaluation techniques substantially different from the current known techniques (FFT and fast matrix multiplication) are used.

The (only) caveat of our lower bound in Theorem 1 is that it relies on using the 0 of R for false (which is very natural when working with $\wedge$ and $\vee$). We leave open the problem of proving a "basis independent" lower bound (where false can correspond to any value in R).

A General Lower Bound for AND and OR

Our main technical tool is a general (basis independent) lower bound for probabilistic polynomials representing the AND and OR functions, which is also of independent interest. The celebrated probabilistic polynomial constructions of Razborov and Smolensky, which are the key component behind the aforementioned probabilistic polynomials in many recent algorithms (including APSP and OV) yield:

▶ **Theorem 2** ([29, 32])**.** *For all primes p and integers $t \geq 1$, OR and AND on n variables have $\mathbb{F}_p$-probabilistic polynomials of degree $(p-1)t$, sparsity $O(n^{(p-1)t})$, and error $1/p^t$.*

Using this construction, there are $\mathbb{F}_p$-probabilistic polynomials for AND on n variables with low values for any *two choices* of the sparsity, degree, and error measures:

- Sparsity $(p-1)n$ and degree $p-1$ (but error $\Omega(1/p)$) follows from Theorem 2 with $t = 1$.
- Degree $\log n$ and error $1/p^{(\log n)/(p-1)}$ (but sparsity $n^{\Omega(\log n)}$) follows from Theorem 2 with $t = (\log n)/(p-1)$.
- Sparsity 1 and error 0 (but degree $\Omega(n)$) is achieved by the exact polynomial $x_1 x_2 \cdots x_n$.

A natural question arises: is there a probabilistic polynomial for AND with low degree, low sparsity, and low error *simultaneously*?

▶ **Question 1.** *Is there a probabilistic polynomial for AND on n variables over any commutative ring which has $O(\log n)$ degree, $1/poly(n)$ error, and $poly(n)$ sparsity?*

While Question 1 is fundamental in its own right, a positive answer would also contradict a non-uniform version of SETH, by using it to construct sparse probabilistic polynomials for OV_c; see the discussion in Appendix B for more details. In constructing these polynomials, there is a choice of which values in our ring R map to true and false. We could use $\{0, 1\}$,

or $\{1, -1\}$, or even a basis like $\{1, 2\}$ [16]. While the degrees of polynomials do not change under such transformations, it is known that in general, the sparsity of polynomials can be very sensitive to this map [21]. Could this be exploited to refute SETH?

We unconditionally prove a strong **no** answer to Question 1:

▶ **Theorem 3.** *For every commutative ring R, every distinct $a, b \in R$, and all $d \geq 1$ and $e > 0$, there is a $c \geq 1$ such that for all sufficiently large m, the AND function on m variables does not have a R-probabilistic polynomial, where a represents false and b represents true, of sparsity $2^{dm/c}$, degree m/c, and error $2^{-em/c}$.*

Indeed, a polynomial giving a yes answer to Question 1, upon setting all but $m = \omega(\log n)$ variables to 1 (or whichever element of the ring R corresponds to true), would contradict Theorem 3, since sparsity, degree, and error cannot increase upon setting variables. We emphasize that Theorem 3 holds over *any* commutative ring R and *any* choice of basis $\{a, b\}$.

Further Results

Our technical lemmas can be applied to yield several sparsity lower bounds for various algebraic representations of natural and well-studied Boolean functions. First, we give an exponential sparsity-error tradeoff lower bound for probabilistic polynomials for the disjointness function $DISJ$ and the "complements" function $COMP$. Informally, $DISJ$ determines whether two given bit-vectors have zero inner product, and $COMP$ determines whether two given bit-vectors are binary complements of each other; it is very closely related to the equality function (see Section 2 for formal definitions).

▶ **Theorem 4.** *For every commutative ring R, function $f : \mathbb{N} \to \mathbb{N}$, constants $e > 0$ and $d \geq 1$, there is a $c \geq 1$ such that the disjointness of two n bit vectors does not have a R-probabilistic polynomial of sparsity $f(c)2^{dn/c}$ and error $f(c)/2^{en/c}$ which uses the 0 of R to represent "false".*

▶ **Theorem 5.** *Theorem 4 also holds with disjointness replaced by $COMP$.*

That is, we cannot obtain a probabilistic polynomial for disjointness with subexponential sparsity and subexponential error simultaneously, *regardless* of the degree. (Observe that the exact representation of disjointness of two n-bit vectors has 2^n sparsity and 0 error.)

In the appendices, we also apply the above results in a number of other ways.

In Appendix A, we give a simple equivalence between probabilistic polynomials and certain depth-3 circuits, which makes some prior work relevant. In Appendix B, we discuss how sparse probabilistic polynomials for simple AC^0 functions would refute SETH:

▶ **Theorem 6** (Informal; see Theorems 24 and 27)**.** *For any commutative ring R, either of the following would refute SETH:*

- *A low-degree, low-sparsity, and low-error R-probabilistic polynomial for AND (over 0/1), or*
- *An R-probabilistic polynomial for $OR_n \circ AND_{c \log n} \circ OR_2$ with sparsity n^k and constant error for any fixed integer k.*

In Appendix C, we investigate probabilistic polynomials over the $\{-1, 1\}$ basis instead of $\{0, 1\}$. While the degree of a polynomial does not depend on the basis, the sparsity can be very sensitive to such a change, and we give examples where probabilistic polynomial sparsity lower bounds in the $\{0, 1\}$ setting do not hold in the $\{-1, 1\}$ setting. Building on our lower bound Theorem 5 over the $\{0, 1\}$ basis, we prove:

▶ **Theorem 7.** *Over any commutative ring R which doesn't have characteristic 2, the complements function $COMP$ has a probabilistic polynomial with sparsity poly(n) and error $1/$poly(n) over the basis $\{-1, 1\}$, and does not over the basis $\{0, 1\}$.*

In Appendix D, we consider a notion related to the probabilistic sparsity of a function, introduced by [4], called the *probabilistic rank*. The probabilistic rank of a function f measures the extent to which the truth table matrix, or communication matrix, of f can be probabilistically represented by low rank matrices. Probabilistic sparsity upper bounds give corresponding probabilistic rank upper bounds, a fact which is used by the aforementioned "polynomial method" algorithms to reduce polynomial evaluation to fast matrix multiplication. For some functions, the probabilistic rank could potentially be much smaller than the probabilistic sparsity. This is interesting, both for the prospect of being able to design faster algorithms using rank instead of sparsity, and given the connection between probabilistic rank and the matrix rigidity problem [4]. In Appendix D, we prove a separation between the two notions in a number of settings over $\mathbb{F}_2$, including showing that the two notions can be arbitrarily far apart for a natural function:

▶ **Theorem 8.** *For any $k > \Omega(n/\log(1/\varepsilon))$, the ε-probabilistic sparsity of $MAJ \circ XOR$ over $\mathbb{F}_2$, where MAJ has fan-in n and each XOR has fan-in k, is strictly greater than its ε-probabilistic rank. In particular, as k increases, the ε-probabilistic sparsity grows unboundedly while ε-probabilistic rank remains fixed.*

We note that there are more straightforward constructions of functions f which separate probabilistic sparsity and probabilistic rank, such as any function of large probabilistic sparsity which depends only on the first half of its input bits and thus has rank 1. However, the functions which are most interesting in algorithmic applications typically depend on both the first and second halves of the input variables similarly.

In Appendix E, we observe how bounds on one of the sparsity or degree of probabilistic polynomials for a function can give bounds on the other as well. We notably show that, if a function requires probabilistic degree d, then it must require probabilistic sparsity $\Omega(2^d)$. Hence, probabilistic degree lower bounds, which are more common in the literature, imply probabilistic sparsity lower bounds as well. We also relate probabilistic degree to probabilistic rank in a better-than-trivial way: Any n-variate Boolean function with a probabilistic polynomial of degree d and error ε has ε-probabilistic rank most $O\left(\binom{n/2}{d/2}\right)$, compared to the trivial $O\left(\binom{n/2}{d}\right)$.

Intuition for our Results

Our main proof technique, which we use to prove Theorem 3, and then again in some of our additional results, is a novel combination of two well-known ideas from the literature on lower bounds for Boolean functions and their polynomial representations.

The first idea is *carefully chosen random restrictions*. Random restrictions are convenient for studying probabilistic polynomial sparsity, since we can study each monomial of a probabilistic polynomial separately, and analyze the distribution of its resulting degree after applying a restriction. We give random restrictions that do not restrict too many variables, but that substantially lower the degree of any sparse polynomial.

The second idea is a variant of the *Schwartz-Zippel Lemma* (Lemma 13 below), which roughly says that low degree multivariate polynomials must have many roots on the Boolean hypercube. This is especially interesting, for instance, when applied to the OR function, which has only one Boolean root; it says that low degree probabilistic polynomials for OR must have high error.

Our proof strategy combines these two ideas in a new way. Given a sparse probabilistic polynomial for AND with low degree and error, we design a random restriction so that the restriction of at least one polynomial in the support of the probabilistic polynomial violates the Schwartz-Zippel Lemma, giving a contradiction. Previous work using random restrictions to get sparsity lower bounds for other types of polynomials has typically combined the random restrictions with simple degree lower bounds. Here, we need some novel technical arguments in order to apply the Schwartz-Zippel Lemma with the trade-off between the error and degree of the resulting restriction.

Of course, our description here (necessarily) glosses over many important details. In addition to random restrictions and the Schwartz-Zippel Lemma, our techniques use multi-party communication complexity, algebraic results about commutative rings (especially in our proof that Theorem 3 holds over any basis), and prior work in the area of polynomial representations of Boolean functions. Our proof extending Theorem 3 to any basis, which we give in Section G, is very general, and could be of independent interest for proving basis-independent versions of other results. It shows that for any function f, a simultaneous lower bound on the degree and sparsity of a probabilistic polynomial for f over any one basis implies a sparsity lower bound over any other basis as well.

Prior Work

Work on particular depth-three circuit size lower bounds (namely, $\mathsf{MAJ} \circ \mathsf{SYM} \circ \mathsf{AND}$ circuits) can be seen as lower bounds on the sparsity of probabilistic polynomials (see Appendix A for an overview of the simple connection). For example:

- Razborov and Wigderson [30] showed that a simple depth-three $\mathsf{AC}^0[2]$ function f requires $\mathsf{MAJ} \circ \mathsf{SYM} \circ \mathsf{AND}$ circuits of size $n^{\Omega(\log n)}$. This implies $f \notin \mathbb{F}_p\text{-SDE}_{0,1}[n^{o(\log n)}, n, 1/2 - \varepsilon]$ for any fixed prime p, a sparsity/degree/error lower bound for probabilistic polynomials computing f.
- Chattopadhyay [12] gave an AC^0 function requiring exponential $\mathsf{MAJ} \circ \mathsf{SYM} \circ \mathsf{ANY}$ circuits when the bottom fan-in is $o(\log \log n)$; this corresponds to an $\Omega(\log \log n)$ degree lower bound for probabilistic polynomials computing an AC^0 function.
- Beame and Huynh [8] give a depth-9 AC^0 circuit that needs $\mathsf{MAJ} \circ \mathsf{SYM} \circ \mathsf{AND}$ circuits of size $n^{\Omega(\log n)}$, corresponding to a probabilistic $n^{\Omega(\log n)}$-sparsity lower bound for this AC^0 function.
- Sherstov [31] gives a depth-3 AC^0 circuit with tighter $\mathsf{MAJ} \circ \mathsf{SYM} \circ \mathsf{AND}$ circuit size lower bound. Essentially, if the bottom fan-in is $(1/2 - \varepsilon) \log n$ for some $\varepsilon > 0$, then the circuit size must be exponential.

Our Lemma 20, which we prove on the way to proving Theorem 1, can be viewed as extending this line of work by showing a probabilistic $n^{\Omega(\log n)}$-sparsity lower bound for a depth-2 AC^0 circuit.

Polynomial sparsity has been studied in a stronger setting, where the polynomials in question only need to correlate with a target function f rather than probabilistically represent f, by Lovett and Srinivasan [24]. Their main result is that polynomials over $\mathbb{F}_2$ with $n^{o(\log n)}$ monomials have exponentially low correlation with the $\mathsf{MOD3}$ function. Note that a sparse probabilistic polynomial for a Boolean function f implies the existence of a sparse polynomial with high correlation with f, but not vice versa. Hence, Lovett and Srinivasan's lower bound applies to probabilistic polynomials as well. Their technique has similarities to some of ours – they also use a type of variable restriction to reduce low sparsity polynomials to low degree

polynomials – but their "tree restriction" technique and analysis isn't powerful enough to apply in our setting where, as we will see, we are constrained in how we are allowed to restrict our variables based on the function we are computing.

Sparsity has also been studied in the setting of polynomial threshold functions, where the sign of a polynomial dictates its output (as a Boolean function). Krause and Pudlak [21, 20] give a function f that has exponentially-high sparsity as a polynomial threshold function in the $\{-1, 1\}$ basis, but polynomially-low sparsity over $\{0, 1\}$. Other references on sparsity in the polynomial threshold function setting include Basu *et al.* [7], O'Donnell and Servedio [28], and Hansen and Poldoskii [16]. It should be noted that these polynomial threshold function sparsity lower bounds *do not suffice* to prove probabilistic polynomial sparsity lower bounds: they correspond to lower bounds against depth-2, $\mathsf{MAJ} \circ \mathsf{AND}$ circuits, compared to our lower bounds against probabilistic polynomials which correspond to depth-3, $\mathsf{MAJ} \circ \mathsf{SYM} \circ \mathsf{AND}$ circuits.

The degrees of probabilistic polynomials are more well-studied, starting with the work of Razborov and Smolensky [29, 32], although even some basic questions remain open. For instance, while AND is known to have constant-degree probabilistic polynomials for constant error over any fixed finite field, it is unknown what degree is needed over $\mathbb{R}$: there is a gap between the best upper bound of $O(\log n)$ [9, 34, 5] and the best lower bound of $\tilde{\Omega}(\sqrt{\log n})$ [25, 17]. Our connection between sparsity and degree which we prove in Appendix E makes use of the probabilistic degree of AND, and is thus only (close to) tight over a finite field.

2 Preliminaries

Definitions and Notation

As usual, all logarithms are assumed to be base-two. For gates $\mathcal{G}_1$ and $\mathcal{G}_2$, we write $\mathcal{G}_1 \circ \mathcal{G}_2$ to denote the circuit whose output gate is a $\mathcal{G}_1$ whose inputs are all copies of $\mathcal{G}_2$ on disjoint variables. We write AND_k, OR_k, and XOR_k to denote the AND, OR, and XOR functions on k inputs, respectively, and MAJ is the majority function which computes whether at least half of its inputs are 1.

Let $\mathcal{F} = \{f_n : \{0,1\}^n \to \{0,1\}\}$ be a decision problem, construed as an infinite family of Boolean functions (one for each $n \in \mathbb{N}$). To facilitate the presentation, we define a (non-uniform) complexity class for problems which are computable by sparse, low-degree, and low-error probabilistic polynomials over R:

▶ **Definition 9.** *A decision problem $\mathcal{F}$ is in the class R-**SDE**$[s(n), d(n), e(n)]$ if for all but finitely many n, there is a probabilistic R-polynomial $\mathcal{P}_n$ for $f_n \in \mathcal{F}$ with* **sparsity** *at most $s(n)$,* **degree** *at most $d(n)$, and* **error** *at most $e(n)$. We analogously define R-**SE**$[s(n), e(n)]$ and R-**DE**$[d(n), e(n)]$ when there is no bound on the third parameter, and R-**SDE**$_{a,b}[s(n), d(n), e(n)]$ for probabilistic polynomials over the basis where "false" means $a \in R$ and "true" means $b \in R$ (when omitted, the default is that $a = 0$ and $b = 1$).*

With this notation, we can succinctly state results about probabilistic polynomials for Boolean functions.

Many of our lower bounds concern two different Boolean functions, the *disjointness function*, and the *complements function*.

▶ **Definition 10.** *For vectors $x, y \in \{0,1\}^n$, the* disjointness function*, $DISJ : \{0,1\}^{2n} \to \{0,1\}$, is given by $DISJ(x_1, \ldots, x_n, y_1, \ldots, y_n) = \bigwedge_{i=1}^{n}(x_i \vee y_i)$. (Note that this definition negates the variables compared to how the disjointness function often appears in the literature.)*

▶ **Definition 11.** *For vectors $y \in \{0,1\}^n$, we write $\bar{y}$ to denote the* complement *of y, given by $\bar{y}_i = 1 - y_i$ for each $1 \leq i \leq n$. The* equality function*, $EQ : \{0,1\}^{2n} \to \{0,1\}$, is given by, for $x, y \in \{0,1\}^n$, $EQ(x,y) = 1$ if $x = y$ and $EQ(x,y) = 0$ otherwise. In other words, $EQ(x_1, \ldots, x_n, y_1, \ldots, y_n) := \bigwedge_{i=1}^n (x_i \oplus y_i \oplus 1)$.*

The complements function*, $COMP : \{0,1\}^{2n} \to \{0,1\}$, is given by, for $x, y \in \{0,1\}^n$, $COMP(x,y) = EQ(x,\bar{y})$. In other words, $COMP(x_1, \ldots, x_n, y_1, \ldots, y_n) := \bigwedge_{i=1}^n (x_i \oplus y_i)$.*

Chernoff Bound

In a few of our proofs, we will use a standard Chernoff bound for sums of independent Bernoulli random variables.

▶ **Lemma 12** (Chernoff bound)**.** *If $X_1, \ldots, X_n$ are independent Bernoulli random variables with sum $X = X_1 + \cdots + X_n$, and μ is the expected value of X, then we have:*

$\Pr[X \leq (1-\delta)\mu] \leq \exp\{-\delta^2\mu/2\}$ *for all* $0 < \delta < 1$*, and*

$\Pr[X \geq (1+\delta)\mu] \leq \exp\{-\delta^2\mu/3\}$ *for all* $0 < \delta$*.*

Useful Facts About Polynomials

We state two key facts from past work which we will use in the proofs of our main results. First, we will need the following variant of the Schwartz-Zippel-DeMillo-Lipton Lemma (which appears, for instance, in [27, Lemma 2.6]).

▶ **Lemma 13** (The 0-1 Schwartz-Zippel Lemma)**.** *Let p be a nonzero multilinear n-variate polynomial over any commutative ring R, of total degree at most d. Then $\Pr_{x \sim \{0,1\}^n}[p(x) \neq 0] \geq 1/2^d$.*

A proof can be found in Appendix G for completeness. Note that Lemma 13 is tight for $p(x_1, \ldots, x_d) = \prod_{i=1}^d x_i$. Second, we need a simple lower bound on the probabilistic degree of $AND \circ OR$ circuits which can be derived from a communication complexity lower bound. We refer the reader to the introduction of Sherstov's paper [31] for the relevant definitions (of the number-on-forehead communication model, and the k-party set disjointness problem), as we will only need the statement of Corollary 15 for this paper.

▶ **Theorem 14** ([31] Theorem 1.1)**.** *The number-on-forehead communication complexity of k-party set disjointness on n elements with error $1/3$ is at least $\Omega(\sqrt{n}/(2^k k))$.*

▶ **Corollary 15.** *For every $\varepsilon > 0$ and every $a > 0$, there is a $c > 0$ such that: For every sufficiently large positive integer n, every positive integer k with $k < c \cdot \log n$, and every finite commutative ring R with $|R| \leq 2^{a \cdot n^{1/2-\varepsilon}}$, the function $AND \circ OR$, where the AND has fan-in n and each OR has fan-in k, does not have a probabilistic polynomial of error $1/3$ and degree less than k.*

Proof (sketch). From such a polynomial, we can design a $1/3$ error number-on-forehead communication protocol for k-party set disjointness: draw a polynomial from the probabilistic polynomial, then map each monomial to a player whose variable is not in that monomial. Each player can evaluate their monomials and report the sum, from which they can compute set disjointness with the desired error. Each player only needs to communicate one element of R, using $O(\log(|R|)) = O(a \cdot n^{1/2-\varepsilon})$ bits of communication, so the total communication is $O(a \cdot k \cdot n^{1/2-\varepsilon})$. If $k < c \cdot \log n$ for sufficiently small $c > 0$, then this contradicts Theorem 14. ◀

3 Sparsity Lower Bounds for Probabilistic Polynomials

We begin by proving a probabilistic polynomial lower bound for AND:

▶ **Theorem 16.** *For every commutative ring R, and all $d \geq 1$ and $e > 0$, there is a $c \geq 1$ such that for all sufficiently large m, the AND function on m variables is not in R-**SDE**$[2^{dm/c}, m/c, 2^{-em/c}]$.*

For notational simplicity, we use the substitution $m = c \log n$ in our proof:

▶ **Restatement of Theorem 16.** *For every commutative ring R, and all $d \geq 1$ and $e > 0$, there is a $c \geq 1$ such that for all sufficiently large n, the AND function on $c \log n$ variables is not in R-**SDE**$[n^d, \log n, 1/n^e]$.*

Theorem 16 is a special case of Theorem 3, restricted to when the basis is $\{0, 1\}$. Theorems 4 and 5 will follow from this. We will later prove the more general Theorem 3, which holds for any basis over R, in Appendix G. In other words, *we are currently focusing on the $\{0, 1\}$ basis, but we later generalize this statement to hold over any basis.* Before we give the proof, let us show that Theorem 16 implies Theorems 4 and 5.

Proof of Theorem 4. Let P be an R-**SE**$[f(c)2^{dn/c}, f(c)/2^{en/c}]$ representation of $\text{AND}_n \circ \text{OR}_2$. From each of the OR_2 gates, pick at random one variable feeding into the gate and fix it 0. Thus, for each c', every monomial in P of degree at least $c'n/c$ survives with probability at most $1/2^{c'n/c}$. The probability that P under this restriction has degree greater than $c'n/c$ is therefore at most $f(c)2^{dn/c} \cdot \frac{1}{2^{c'n/c}}$. We set $c > c' > d$.

Note that under this restriction, the disjointness function reduces to AND_n. Thus, we have constructed a probabilistic polynomial for AND_n of degree $c'n/c$ with error $\frac{f(c)2^{dn/c}}{2^{c'n/c}} + \frac{f(c)}{2^{en/c}}$. If c is large enough, this contradicts Theorem 16. Hence our assumption about the existence of P must be false. ◀

Proof of Theorem 5. The proof is identical, since restricting one input of each XOR_2 gate to 0 reduces $COMP$ to AND as well. ◀

We now begin proving Theorem 16. We start with a simple case to illustrate our use of the 0-1 Schwartz-Zippel Lemma (Lemma 13):

▶ **Lemma 17.** *For every commutative ring R, and all $e > 1$, there is a $c > 1$ such that for all sufficiently large n, the AND function on $c \log n$ variables is not in R-**DE**$[\log n, 1/n^e]$.*

Proof. Let $\mathcal{P}$ be a probabilistic R-polynomial with error at most $1/n^e$ computing AND of $c \log n$ bits, such that every polynomial in the distribution has at most $\log n$ degree.

First we claim that, without loss of generality, the support of $\mathcal{P}$ may be assumed to not contain the identically zero polynomial z. Since $\mathcal{P}$ is a polynomial for AND with error at most $1/n^e$, on the point $y = (1, \ldots, 1)$ we must have $\Pr_{p \sim \mathcal{P}}[p(y) = 1] \geq 1 - 1/n^e$. Therefore, if z is in the support of $\mathcal{P}$, it must have probability at most $1/n^e$ of being chosen. Hence if we simply replace z in $\mathcal{P}$ with the polynomial which is identically the constant 1, the new probability of error is at most $2/n^e$.

Next, we prove there is no distribution $\mathcal{P}$ satisfying the above properties, if $e > 1$. Otherwise, by fixing randomness in $\mathcal{P}$ (i.e., by picking any polynomial in the support of $\mathcal{P}$ which achieves at most the average error), we can identify a fixed polynomial p of degree at most $\log n$ which disagrees with the AND function on at most a $1/n^e < 1/n$ fraction of inputs. By the previous paragraph, p is *not identically zero*. Since the AND of $c \log n$ variables is nonzero on only one point in $\{0, 1\}^{c \log n}$, it follows that the polynomial p is nonzero on at most $n^c/n^e + 1$ points in $\{0, 1\}^{c \log n}$. Hence the nonzero degree-$\log n$ polynomial p satisfies

$$\Pr_{x \sim \{0,1\}^{c \log n}}[p(x) \neq 0] \leq 1/n^e + 1/n^c,$$

contradicting Lemma 13 when $e, c > 1$. ◀

Before we turn to the harder part of the proof, where $e \leq 1$, we sketch the main ideas. Our first step is to hit a presumed probabilistic polynomial for AND with a random restriction of 1-inputs. The probabilities are chosen carefully so that the degree of the polynomial decreases considerably, yet the number of variables restricted is not too small, with high probability. That is, we shrink the degree of the polynomial, but we do not force it to a constant. Next, we select a non-zero polynomial from the remaining probabilistic polynomial distribution, and argue that with high probability, its degree has dropped significantly below $e \log n$, but its error on the AND function remains about $O(1/n^e)$. This contradicts Lemma 13, which says that a non-zero polynomial of degree-d must be non-zero on at least $1/2^d$ points. While the general idea of the proof is not too complicated, we need to be careful with the details, and choose parameters very particularly so that all of the required constraints hold. Let us now give the details:

Proof of Theorem 16. Assume there is a probabilistic polynomial $\mathcal{P}$ over R for AND with $d, e > 0$ such that for all $c \geq 1$, the sparsity is at most n^d, the degree is $\log n$, and the error is $1/n^e$. As in the proof of Lemma 17, we may assume that the zero polynomial is not in the support of $\mathcal{P}$. Let $\alpha, \beta, \gamma > 0$ be real-valued parameters to be set later. Define

$$g := (1+\alpha)(\log n)/\gamma, \text{ and } \ell := (1-\beta)(c \log n)/\gamma.$$

Consider the following construction $\mathcal{Q}$ of a probabilistic polynomial for AND on ℓ bits:

- Given $(x_1, \ldots, x_\ell) \in \{0,1\}^\ell$, sample $p(y_1, \ldots, y_{c \log n}) \sim \mathcal{P}$.
- For $i = 1, \ldots, c \log n$, independently and with probability $1 - 1/\gamma$, assign y_i to 1 in the polynomial p. Let p' be the remaining polynomial in $\text{vars}(p') \leq c \log n$ variables.
- If $\text{vars}(p') \geq \ell$ and $\deg(p') \leq g$, then output $p'(x_1, \ldots, x_\ell, 1, \cdots, 1)$ (where the $\cdots$ indicate $\text{vars}(p') - \ell$ ones).
- Otherwise, output a random bit.

Our central claim is that, for appropriate setting of α, β, γ, the probabilistic polynomial $\mathcal{Q}$ has degree at most g and computes the AND of ℓ bits with error at most $O(1/n^e)$. The degree of $\mathcal{Q}$ and AND functionality follow by construction; we need to bound the error. This will follow from positing a series of inequalities in the analysis that imply a small error bound, then proving at the end that the inequalities can be satisfied.

Define $\text{err} := \Pr[(\deg(p') > g) \vee (\text{vars}(p') < \ell)]$, where the probability is over the sampling in $\mathcal{P}$ (of step 1) and the random restriction to p' (of step 2). Observe that err is precisely the probability that case 4 is reached in the above procedure. Further observe that the error of $\mathcal{Q}$ is at most $1/n^e + \text{err}$: we have $1/n^e$ probability of error from $\mathcal{P}$ which is $1/n^e$, and probability err of reaching case 4.

We now claim that $\text{err} \leq 2/n^e$, when $\alpha, \beta, \gamma > 0$ are set appropriately. In particular, provided that

$$\frac{\alpha^2(1+\alpha)}{3\gamma^2} \geq d + e, \quad \text{and} \tag{1}$$

$$\frac{\beta^2 c}{2\ln(2)\gamma} \geq e, \tag{2}$$

we will have that both $\Pr[\deg(p') > g]$ and $\Pr[\text{vars}(p') < \ell]$ are at most $1/n^e$.

Observe that the number of variables $\text{vars}(p')$ in p' can be seen as a sum of independent $0-1$ random variables $Y_1, \ldots, Y_{c \log n}$, such that $\Pr[Y_i] = 1/\gamma$. Hence $\text{vars}(p')$ has expectation $(c \log n)/\gamma$ and by a Chernoff bound (Lemma 12),

$$\Pr[\text{vars}(p') < \ell] = \Pr[\sum_i Y_i < (1-\beta)(c \log n)/\gamma] \leq \exp\{-\beta^2 (c \log n)/(2\gamma)\}.$$

But if (2) holds, then $\Pr[\text{vars}(p') < \ell] < \exp\{-\beta^2 (c \log n)/(2\gamma)\} \leq \exp\{-e \ln n\} = 1/n^e$.

We will analyze $\Pr[\deg(p') > g]$ by considering the monomials of p. Let m be a monomial in $p \sim \mathcal{P}$ of degree $\deg(m) \leq \log n$. If $\deg(m) \leq g$, then the monomial m cannot possibly contribute to the event that $\deg(p') > g$ (the degree cannot increase by setting variables). So we may assume $\deg(m) > g$. There is a corresponding randomly reduced monomial m' in p', obtained by setting each of the variables in monomial m to 1 with probability $1/\gamma$. (Note this is the worst case; in some commutative rings R, the monomial m' could also cancel with another monomial and disappear from p'.) The degree $\deg(m')$ can be seen as a sum of independent 0-1 random variables $Z_1, \ldots, Z_{\deg(m)}$ where $\Pr[Z_i] = 1/\gamma$. By a Chernoff bound (Lemma 12),

$$\begin{aligned}
\Pr[\deg(m') > (1+\alpha)\deg(m)/\gamma] &= \Pr[\sum_i Z_i > (1+\alpha)\deg(m)/\gamma] \\
&\leq \exp\{-\alpha^2 \deg(m)/(3\gamma)\} \\
&< \exp\{-\alpha^2 g/(3\gamma)\} = \exp\{-\alpha^2(1+\alpha)(\log n)/(3\gamma^2)\}.
\end{aligned}$$

Now, if (1) holds, then $(\alpha^2(1+\alpha))/(3\ln(2)\gamma^2) \geq d+e$ and therefore

$$\Pr[\deg(m') > (1+\alpha)\deg(m)/\gamma] < \exp\{-\alpha^2(1+\alpha)(\log n)/(3\gamma^2)\} \leq \exp\{-(d+e)\ln n\} = 1/n^{d+e}.$$

This bound and $\deg(m) \leq \log n$ holds for every monomial m in p, and there are at most n^d monomials, so

$$\begin{aligned}
\Pr[\deg(p') > (1+\alpha)(\log n)/\gamma] &\leq \Pr\left[(\exists \text{ monomial } m')[\deg(m') > (1+\alpha)(\log n)/\gamma]\right] \\
&\leq \Pr\left[(\exists \text{ monomial } m')[\deg(m') > (1+\alpha)\deg(m)/\gamma]\right] \\
&\leq n^d/n^{d+e} = 1/n^e,
\end{aligned}$$

by the union bound.

We have proved that, assuming α, β, γ are set properly, $\mathcal{Q}$ has degree at most g and computes the AND of ℓ bits with error at most $O(1/n^e)$. Finally, suppose that

$$\frac{1+\alpha}{\gamma} < e. \tag{3}$$

Then, $\mathcal{Q}$ is a probabilistic polynomial for the AND on ℓ variables, with degree at most $g = (1+\alpha)(\log n)/\gamma < e \log n$ and error $O(1/n^e)$.

Fixing the randomness in $\mathcal{Q}$, we are in an analogous situation as the earlier case of $e > 1$: we can find a single non-zero degree-g polynomial q that differs from the AND of ℓ variables on an $O(1/n^e)$-fraction of points. Since the AND is nonzero on a $1/2^\ell$ fraction of points, the polynomial q is non-zero on at most an $O(1/n^e) + 1/2^\ell$ fraction of points. Provided that

$$\frac{(1-\beta)c}{\gamma} > e, \tag{4}$$

we will have $\ell = (1-\beta)(c \log n)/\gamma > e \log n$ and $1/n^e > 1/2^\ell$, hence q is non-zero on an $O(1/n^e)$ fraction of points. But then the degree of q is $\deg(q) \leq g \leq (e-\varepsilon)\log n$, for some $\varepsilon > 0$. This contradicts Lemma 13.

It remains to show that (1), (2), (3), and (4) can be simultaneously satisfied to yield the above contradiction, which we do in Lemma 46 in Appendix F. ◀

The full proof of Theorem 3, which generalizes the above Theorem 16 to any basis $\{a, b\}$ over R, can be found in Appendix G. It shows that picking representatives in R for true and false other than 1 and 0 cannot help to overcome the lower bound of Theorem 16. It is unfortunately not so straightforward to prove from here, since our proof of Theorem 16 critically uses the Schwartz-Zippel lemma, Lemma 13, which does not hold for general a, b over any commutative ring R. For a simple example, if $(a, b) = (0, 3)$ over $R = \mathbb{Z}/6\mathbb{Z}$, then the nonzero linear polynomial $p(x) = 2x$ is nonetheless zero on both a and b. We solve this issue by taking any probabilistic polynomial $\mathcal{P}$ over R with a basis $\{a, b\}$ where the Schwartz-Zippel lemma does not hold, and performing a combinatorial transformation to yield a new probabilistic polynomial $\mathcal{Q}$ over R with a different basis where the Schwartz-Zippel lemma does hold, and such that $\mathcal{Q}$ has the same degree and error, and only mildly greater sparsity, than $\mathcal{P}$.

3.1 Probabilistic Sparsity Lower Bounds for Compositions

We next show how probabilistic degree lower bounds for Boolean functions f can lead to probabilistic sparsity lower bounds for compositions of f with simple functions like OR and XOR.

▶ **Theorem 18.** *Let R be any commutative ring, and let $f : \{0,1\}^n \to \{0,1\}$ be any Boolean function such that any probabilistic polynomial of error ε for f over R requires degree d. Then, for every $0 < a < \varepsilon$, every $(\varepsilon - a)$-error probabilistic polynomial over R for the kn-variate function $f \circ OR_k$, the composition of f with ORs of k disjoint variables, requires sparsity at least $a \cdot k^d$ over the $\{0,1\}$ basis.*

Proof. Label the variables of our function of interest, $f \circ OR_k$, by $x_{i,j}$ for $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, k\}$, so that our function can be written $f\left(\bigvee_{j=1}^{k} x_{1,j}, \ldots, \bigvee_{j=1}^{k} x_{n,j}\right)$. Consider the following random restriction ρ: for each $i = 1, \ldots, n$, choose one $j \in \{1, \ldots, k\}$ uniformly and independently at random, and set $x_{i,j'}$ to 0 for each $j' \neq j$. Notice in particular that after applying ρ to the inputs of our function $f \circ OR_k$, the result is the function f on the remaining n variables which were not set to 0.

Assume to the contrary that a probabilistic polynomial P exists for $f \circ OR_k$ with sparsity less than $a \cdot k^d$. We construct a probabilistic polynomial for f as follows:

1. Draw a $p \sim P$, and a random restriction ρ as described above.
2. Let p_ρ be the polynomial obtained by substituting 0s chosen by ρ into the variables of p.
3. If $deg(p_\rho) < d$ then output p_ρ, otherwise output a random bit.

Since the restriction ρ transforms $f \circ OR_k$ into f, we know that p_ρ as constructed in step 2 is a probabilistic polynomial for f with error $(\varepsilon - a)$. If we can show that we actually return p_ρ with probability at least $1 - a$ in step 3, then we know our probabilistic polynomial has error at most $(\varepsilon - a) + a = \varepsilon$, and it by definition has degree less than d, which will contradict our assumption about f as desired.

Observe that the monomials of p_ρ are a subset of the monomials of p. Moreover, a monomial of p appears in p_ρ if and only if none of its variables were set to 0. For a monomial m in p of degree at least d, the probability that m appears in p_ρ is at most $1/k^d$: if both x_{i,j_1} and x_{i,j_2} appear in m for some i and $j_1 \neq j_2$, then m is definitely set to zero upon restriction ρ, and otherwise, each variable in m is set to 0 with probability $1 - 1/k$ independently of the rest.

Therefore, if p has sparsity $|p|$, then the expected number of monomials of degree at least d in p_ρ is at most $|p|/k^d \leq a$. Hence, by Markov's inequality, the probability that p_ρ has degree at least d, meaning it has at least one monomial of degree at least d, is at most a, as desired. ◀

▶ **Corollary 19.** *Theorem 18 also holds with $f \circ OR_k$ replaced by $f \circ XOR_k$.*

Applying Theorem 18 and Corollary 15, we can prove that a simple depth-2 AC^0 circuit requires probabilistic sparsity $n^{\Omega(\log n)}$. (Note the lower bound is tight, as mentioned in the introduction.)

▶ **Lemma 20.** *For every $\varepsilon > 0$ and every finite commutative ring R such that $|R| \leq 2^{O(n^{1/2-\varepsilon})}$, and every function $g : \mathbb{N} \to \mathbb{N}$, R-probabilistic polynomials with error $1/4$ for $AND \circ OR$, where the AND has fan-in n and each OR has fan-in $g(n) \cdot \log(n)$, require sparsity $n^{\Omega(\log g(n))}$ over the $\{0,1\}$ basis.*

Proof. Pick a sufficiently small $c > 0$ such that Corollary 15 holds, meaning $f := AND_n \circ OR_{c \log n}$ requires R-probabilistic degree $\Omega(\log n)$ for error $1/3$. Applying Theorem 18 to f with $a = 1/12$ and $k = g(n)$, we see that $f \circ OR_{g(n)}$ requires $\mathbb{F}_2$-probabilistic sparsity $(g(n))^{\Omega(\log n)} = n^{\Omega(\log g(n))}$ for error $1/4$. Collapsing the two bottom layers of ORs, we see that $f \circ OR_{g(n)} = AND_n \circ OR_{c \cdot g(n) \log n}$, which implies the desired result since $c \cdot g(n) \log n < g(n) \log(n)$. ◀

We can finally prove our main theorem:

Proof of Theorem 1. Setting $y_{i,j} = 1$ for all i, j changes OV_g into the negation of the function from Lemma 20. The same lower bounds hold since setting variables does not increase the degree, sparsity, or error. ◀

4 Conclusion

By combining multiple techniques (random restrictions, Schwartz-Zippel, communication complexity lower bounds, known degree lower bounds, etc.), we have shown sparsity lower bounds for probabilistic and approximate polynomials computing several natural functions of interest in algorithm design. Perhaps the main question left open by our work is whether there is a polynomially-sparse constant-error probabilistic polynomial for the depth-three $OR_n \circ AND_n \circ OR_2$ circuit over *some* basis $\{a, b\}$ where $a \neq b$, working in *some* ring R. Even more generally, does this circuit have low *probabilistic rank* over some ring? In this paper, we proved several strong sparsity lower bounds over the basis $\{0, 1\}$ which significantly narrows the search space for such polynomials, but some of steps in our proofs require that the basis contains 0 ("killing" monomials by setting variables to 0).

For the complements ($COMP$) function, there is a nice sparsity upper bound over $\{-1, 1\}$ (compared to the lower bound over $\{0, 1\}$), so it is still possible (but unlikely) that the $\{-1, 1\}$ basis could support sparse probabilistic polynomials for depth-3 AC^0 functions. More basis-independent methods for sparsity lower bounds (such as Theorem 3) would be of great interest, as well as new techniques for constructing sparse polynomial representations.

References

1 Amir Abboud, Ryan Williams, and Huacheng Yu. More applications of the polynomial method to algorithm design. In *SODA*, pages 218–230, 2015.

2 Josh Alman, Timothy M Chan, and Ryan Williams. Polynomial representations of threshold functions and algorithmic applications. In *FOCS*, pages 467–476, 2016. doi:10.1109/FOCS.2016.57.

3 Josh Alman and Ryan Williams. Probabilistic polynomials and hamming nearest neighbors. In *FOCS*, pages 136–150, 2015. doi:10.1109/FOCS.2015.18.

4 Josh Alman and Ryan Williams. Probabilistic rank and matrix rigidity. In *STOC*, pages 641–652, 2017. doi:10.1145/3055399.3055484.

5 James Aspnes, Richard Beigel, Merrick Furst, and Steven Rudich. The expressive power of voting polynomials. *Combinatorica*, 14(2):135–148, 1994. doi:10.1007/BF01215346.

6 David A Mix Barrington, Richard Beigel, and Steven Rudich. Representing boolean functions as polynomials modulo composite numbers. *Computational Complexity*, 4(4):367–382, 1994. doi:10.1007/BF01263424.

7 Saugata Basu, Nayantara Bhatnagar, Parikshit Gopalan, and Richard J. Lipton. Polynomials that sign represent parity and descartes' rule of signs. *Computational Complexity*, 17(3):377–406, 2008. doi:10.1007/S00037-008-0244-2.

8 Paul Beame and Trinh Huynh. Multiparty communication complexity and threshold circuit size of AC^0. *SIAM Journal on Computing*, 41(3):484–518, 2012. doi:10.1137/100792779.

9 Richard Beigel, Nick Reingold, and Daniel Spielman. The perceptron strikes back. In *Structure in Complexity Theory Conference*, pages 286–291. IEEE, 1991. doi:10.1109/SCT.1991.160270.

10 Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. The complexity of satisfiability of small depth circuits. In *Parameterized and Exact Complexity (IWPEC)*, pages 75–85, 2009. doi:10.1007/978-3-642-11269-0_6.

11 Timothy M Chan and Ryan Williams. Deterministic apsp, orthogonal vectors, and more: Quickly derandomizing razborov-smolensky. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1246–1255. Society for Industrial and Applied Mathematics, 2016. doi:10.1137/1.9781611974331.CH87.

12 Arkadev Chattopadhyay. Discrepancy and the power of bottom fan-in in depth-three circuits. In *FOCS*, pages 449–458. IEEE, 2007. doi:10.1109/FOCS.2007.30.

13 Ernie Croot, Vsevolod F Lev, and Péter Pál Pach. Progression-free sets in Zn. *Annals of Mathematics*, 185:331–337, 2017.

14 Zeev Dvir and Benjamin L Edelman. Matrix rigidity and the croot-lev-pach lemma. *Theory Of Computing*, 15(8):1–7, 2019. doi:10.4086/TOC.2019.V015A008.

15 Jiawei Gao, Russell Impagliazzo, Antonina Kolokolova, and R. Ryan Williams. Completeness for first-order properties on sparse structures with algorithmic applications. In *SODA*, pages 2162–2181, 2017. doi:10.1137/1.9781611974782.141.

16 Kristoffer Arnsfelt Hansen and Vladimir V Podolskii. Polynomial threshold functions and boolean threshold circuits. *Information and Computation*, 240:56–73, 2015. doi:10.1016/J.IC.2014.09.008.

17 Prahladh Harsha and Srikanth Srinivasan. On polynomial approximations to ac. *Random Structures & Algorithms*, 54(2):289–303, 2019. doi:10.1002/RSA.20786.

18 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *JCSS*, 62(2):367–375, 2001. doi:10.1006/JCSS.2000.1727.

19 Swastik Kopparty and Srikanth Srinivasan. Certifying polynomials for AC^0(parity) circuits, with applications. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012*, pages 36–47, 2012. doi:10.4230/LIPICS.FSTTCS.2012.36.

20 Matthias Krause and Pavel Pudlák. On the computational power of depth-2 circuits with threshold and modulo gates. *Theoretical Computer Science*, 174(1-2):137–156, 1997. doi:10.1016/S0304-3975(96)00019-9.

21 Matthias Krause and Pavel Pudlák. Computing boolean functions by polynomials and threshold circuits. *computational complexity*, 7(4):346–370, 1998. doi:10.1007/S000370050015.

22 Kasper Green Larsen and R. Ryan Williams. Faster online matrix-vector multiplication. In *SODA*, pages 2182–2189, 2017. doi:10.1137/1.9781611974782.142.

23 Daniel Lokshtanov, Ramamohan Paturi, Suguru Tamaki, R. Ryan Williams, and Huacheng Yu. Beating brute force for systems of polynomial equations over finite fields. In *SODA*, pages 2190–2202, 2017. doi:10.1137/1.9781611974782.143.

24 Shachar Lovett and Srikanth Srinivasan. Correlation bounds for poly-size AC^0 circuits with $n^{1-o(1)}$ symmetric gates. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 640–651. Springer, 2011.

25 Raghu Meka, Oanh Nguyen, and Van Vu. Anti-concentration for polynomials of independent random variables. *arXiv preprint*, 2015. arXiv:1507.00829.

26 Daniel Moeller, Ramamohan Paturi, and Stefan Schneider. Subquadratic algorithms for succinct stable matching. In *Computer Science Symposium in Russia*, pages 294–308, 2016. doi:10.1007/978-3-319-34171-2_21.

27 Noam Nisan and Mario Szegedy. On the degree of boolean functions as real polynomials. *Computational complexity*, 4(4):301–313, 1994. doi:10.1007/BF01263419.

28 Ryan O'Donnell and Rocco A Servedio. Extremal properties of polynomial threshold functions. In *IEEE Conference on Computational Complexity*, pages 3–12. Citeseer, 2003. doi:10.1109/CCC.2003.1214406.

29 A. A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical Notes of the Academy of Sciences of the USSR*, 41(4):333–338, 1987.

30 Alexander Razborov and Avi Wigderson. $n^{\Omega(\log n)}$ lower bounds on the size of depth-3 threshold cicuits with and gates at the bottom. *Information Processing Letters*, 45(6):303–307, 1993. doi:10.1016/0020-0190(93)90041-7.

31 Alexander A Sherstov. Communication lower bounds using directional derivatives. *Journal of the ACM (JACM)*, 61(6):34, 2014.

32 Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *STOC*, pages 77–82, 1987. doi:10.1145/28395.28404.

33 Srikanth Srinivasan. On improved degree lower bounds for polynomial approximation. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS*, pages 201–212, 2013. doi:10.4230/LIPICS.FSTTCS.2013.201.

34 Jun Tarui. Probabilistic polynomials, ac0 functions and the polynomial-time hierarchy. *Theoretical computer science*, 113(1):167–183, 1993. doi:10.1016/0304-3975(93)90214-E.

35 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoretical Computer Science*, 348(2-3):357–365, 2005. doi:10.1016/J.TCS.2005.09.023.

36 Ryan Williams. Faster all-pairs shortest paths via circuit complexity. In *STOC*, pages 664–673, 2014. doi:10.1145/2591796.2591811.

37 Ryan Williams. The polynomial method in circuit complexity applied to algorithm design (invited talk). In *34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS*, pages 47–60, 2014.

38 Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the ICM*, 2018.

A Depth-3 Circuits and Probabilistic Polynomials

The study of probabilistic polynomials over $\mathbb{F}_p$ is equivalent to studying depth-three circuits of a certain form; let us recall the simple connection. In the following, let $||x||$ denote the number of 1s in the binary string x.

▶ **Definition 21.** *The* approximate majority of separation δ on n bits, *denoted* MAJ_δ, *is defined only on* $x \in \{0,1\}^n$ *with* $||x|| \notin ((1/2-\delta)n, (1/2+\delta)n)$, *and is given by* $MAJ_\delta(x) = 1$ *when* $||x|| \geq (1/2+\delta)n$, *and* $f(x) = 0$ *when* $||x|| \leq (1/2-\delta)n$.

▶ **Lemma 22.** *If f is computable by a depth-three circuit with an approximate majority of separation δ at the output, PARITY gates with fan-in at most s on the middle layer, and AND gates of fan-in at most d at the bottom layer, then there is an $\mathbb{F}_2$ probabilistic polynomial for f with degree d, sparsity s, and error δ.*

Proof. An $\mathbb{F}_2$ polynomial can be viewed as a $PARITY \circ AND$ circuit, and vice versa: each monomial in such a polynomial is computing the AND of its constituent variables, and then the output of the polynomial is the $\mathbb{F}_2$ sum of these monomials, which is a $PARITY$. We thus get our probabilistic polynomial by selecting a uniformly random $PARITY \circ AND$ subcircuit which feeds into the top MAJ_δ gate of our circuit, and outputting it as an $\mathbb{F}_2$ polynomial. ◀

▶ **Lemma 23.** *For any constant $c > 0$, if there is an $\mathbb{F}_2$ probabilistic polynomial for f with degree d, sparsity s, and error δ, then f is computable by a depth-three circuit with an approximate majority of separation $(1-c)\delta$ and fan-in $O(n/\delta)$ at the output, PARITY gates with fan-in at most s on the middle layer, and AND gates of fan-in at most d at the bottom layer.*

If we didn't bound the fan-in of the approximate majority gate, then the proof would be almost identical to the proof of Lemma 22. In order to bound the fan-in by $O(n)$, we need to use a Chernoff bound:

Proof. Randomly sample t polynomials $p_1, \ldots, p_t$ from the probabilistic polynomial, for a parameter t to be selected later. For each $x \in \{0,1\}^n$, by the Chernoff bound, the probability that less than a $(1-c)\delta$ fraction of the polynomials $p_1, \ldots, p_t$ output the correct answer on x is at most $exp(-c^2\delta t)$. By setting $t = a \cdot n/(c^2\delta)$ for a sufficiently large constant a, this can be made less than 2^{-n}. Then, by the union bound, there is a probability less than 1 that, for any $x \in \{0,1\}^n$, less than a $(1-c)\delta$ fraction of the polynomials $p_1, \ldots, p_t$ output the correct answer on x. Hence, by the probabilistic method, there must be a choice of $p_1, \ldots, p_t$ such that at least a $(1-c)\delta$ fraction give the correct answer on every x. We can convert these polynomials into the desired circuit as in the proof of Theorem 22. ◀

It is interesting to note, via Lemmas 22 and 23, that we may assume that any probabilistic polynomial with constant error over $\mathbb{F}_2$ is the uniform distribution on only $O(n)$ different polynomials.

B SETH Predictions about Probabilistic Polynomials

Several algorithms based on the polynomial method in circuit complexity (such as [36, 37, 1, 3, 2]) have the following form: (a) we identify a "subcircuit" C which, if C could be evaluated on many inputs rapidly, we could solve a desired problem faster, (b) efficiently convert C into a polynomial (probabilistic, approximate, etc.) so that the evaluation task becomes algebraic, and (c) solve the evaluation task rapidly using algebraic algorithms, such as a fast Fourier transform or a matrix multiplication. How far can this approach be pushed? A significant question has been whether this kind of approach can be used to solve CNF-SAT fast enough to refute the pesky Strong Exponential Time Hypothesis (SETH).

We observe in this section that a low-degree, low-sparsity, and low-error probabilistic polynomial for AND would have contradicted SETH (recall that we unconditionally prove in this paper that there is no such polynomial). Along the way, we demonstrate other "predictions" about polynomial representations obtained by assuming SETH.

▶ **Theorem 24.** *For any commutative ring R, a low-degree, low-sparsity, and low-error R-probabilistic polynomial for AND (over 0/1) implies that there is a universal $\varepsilon > 0$ such that for all $k \geq 3$, there is a non-uniform circuit family of size $2^{(1-\varepsilon)n}$ for solving k-SAT on n variables.*[1]

The proof of Theorem 24, follows from a few claims. First, we note that a low-degree/low-sparsity/low-error probabilistic polynomial for AND over 0/1 is equivalent to low-sparsity/low-error probabilistic polynomial for $DISJ$ over 0/1. Let R be any commutative ring.

▶ **Proposition 25.** *For any $n, s, d, e \geq 0$, if $AND_n \in R$-**SDE**$[s, d, e]$ then $AND_n \circ OR_2 \in R$-**SE**$[s \cdot 3^d, e]$.*

Proof. From a probabilistic polynomial for AND_n, substitute the exact OR of two variables, $OR(x, y) = x + y - xy$, to get a probabilistic polynomial of the same error for $AND_n \circ OR_2$. In this substitution, each original monomial is replaced by a product of at most d polynomials, each of which is the OR of two variables and has three monomials; so the new polynomial has at most $s \cdot 3^d$ monomials. ◀

Next, we note that a probabilistic polynomial for disjointness ($AND_n \circ OR_2$) with low sparsity and error implies the probabilistic polynomial needed for refuting SETH:

▶ **Theorem 26.** *For any $n, s \geq 0$, if $AND_{c \log n} \circ OR_2 \in R$-**SE**$[s, 1/(8n)]$ then $OR_n \circ AND_{c \log n} \circ OR_2$ has a R-probabilistic polynomial $\mathcal{P}$ of sparsity $O(n^2 \cdot s^2)$ and error 11/24 which (when there is no error) outputs 0 for false and a nonzero value for true.*

Proof. We define $\mathcal{P}$. Let $\mathcal{Q}$ be the R-**SE**$[s, 1/(8n)]$ probabilistic polynomial for $AND_{c \log n} \circ OR_2$. Draw a $q \sim \mathcal{Q}$. By a union bound, with probability at least 7/8, q correctly computes the 0 or 1 value of $AND_{c \log n} \circ OR_2$ on the n different inputs which we want to compute the OR of. Hence, there are n polynomials $q_1, \ldots, q_n$ which each have at most s monomials, and we want to compute their OR assuming they all evalutate to 0 or 1.

The subring R' of R generated by 1 consists of all integer multiples of 1. It is isomorphic to $\mathbb{Z}/m\mathbb{Z}$ for some nonnegative integer $m \neq 1$. (The $m = 0$ case is when no positive integer multiple of 1 is 0 in R, and hence $R' = \mathbb{Z}$.) We consider three cases depending on whether $m = 0$, $m = 2$, or $m > 2$.

If $m = 0$, then simply $q_1 + \cdots + q_n$ suffices to compute OR over $\mathbb{Z}$. This has sparsity at most $n \cdot s$, and no additional error, for a total error of 1/8.

If $m = 2$, then let $S_1, S_2 \subseteq \{1, \ldots, n\}$ be two uniformly random subsets, and we pick

$$-1 + \prod_{i=1}^{2} \left(1 + \sum_{j \in S_i} q_j\right).$$

This is 0 when all the q_j are 0, and each $\sum_{j \in S_i} q_j$ is 1 with probability 1/2 otherwise, so the entire polynomial is 1 with probability at least 3/4. Hence the total error is 3/8, and the sparsity is $O(n^2 \cdot s^2)$.

If $m > 2$, then pick uniformly and independently random $a_1, \ldots, a_n \in \{1, \ldots, m\}$, and pick the polynomial $\sum_{i=1}^{n} a_i q_i$. If each $q_i = 0$ then this is 0, and otherwise it is 0 with probability at most $1/m$. The sparsity is at most $n \cdot s$, and the error is at most $1/8 + 1/m \leq 11/24$. ◀

[1] Moreover, an efficiently-samplable probabilistic polynomial for AND with these properties would refute the original Strong Exponential Time Hypothesis. All interesting probabilistic polynomial constructions we are aware of, such as those found in the works [29, 32, 9, 5, 19, 33, 3, 2], are efficiently samplable.

Finally, we note that a $n^{O(1)}$-sparse constant-error probabilistic polynomial (in the sense of Theorem 26) for $OR_n \circ AND_{c \log n} \circ OR_2$ for *every* $c \geq 1$ would refute (non-uniform) SETH:

▶ **Theorem 27.** *Suppose there is a fixed $k \geq 1$ such that for all $c \geq 1$, $OR_n \circ AND_{c \log n} \circ OR_2 \in R\text{-}\mathbf{SE}[n^k, 11/24]$. Then there is a universal $\varepsilon > 0$ such that for all $k \geq 3$, there is a non-uniform circuit family of size $2^{(1-\varepsilon)n}$ for solving k-SAT on n variables.*

Proof (sketch). The Orthogonal Vectors problem asks: *given a collection of bit vectors, is there a pair of them which are orthogonal?* We show that the hypothesis implies that for all $c \geq 1$, the Orthogonal Vectors problem with n vectors from $\{0,1\}^{c \log n}$ can be solved in $n^{2-\delta}$ time with $O(n^{2-\delta})$ advice, for a universal $\delta > 0$. The theorem then follows from a (by now) standard reduction (as in [35, 3]).

By our hypothesis, we can store (as non-uniform advice) a probabilistic polynomial $\mathcal{P}$ with sparsity $n^{\alpha k}$ and error $11/24$ for the function $OR_{n^\alpha} \circ AND_{c \log n} \circ OR_2$, where $\alpha \in (0, 1/k)$ is a sufficiently small constant. Note this requires storing a distribution of only $O(n)$ distinct $n^{\alpha k}$-sparse polynomials (see Section A).

To solve Orthogonal Vectors on n vectors, we proceed just as in Abboud-Williams-Yu [1]: partition the set of vectors into $O(n^{1-\alpha/2})$ parts, with each part containing at most $n^{\alpha/2}$ vectors. For all $O(n^{2-\alpha})$ pairs (P_1, P_2) of parts, we want to evaluate the probabilistic polynomial $\mathcal{P}$ on all pairs of vectors in the union $P_1 \cup P_2$ of the two parts. This gives an input of n^α pairs of vectors; feeding this input to a polynomial $p \sim \mathcal{P}$, we can estimate whether there is an orthogonal pair among vectors in $P_1 \cup P_2$.

The problem of solving Orthogonal Vectors now reduces to: *evaluate a random $p \sim \mathcal{P}$ with sparsity $n^{\alpha k}$, on $O(n^{2-\alpha})$ different pairs of input vectors.* When $\alpha k < 0.3$ (for example), this problem can be solved in $n^{2-\alpha+o(1)}$ time, via fast rectangular matrix multiplication. To have a high probability of success, we only need to sample $O(\log n)$ random p from our probabilistic polynomial $\mathcal{P}$. ◀

For similar reasons, many types of low sparsity polynomial representations for depth-3 AC^0, including low sparsity approximate polynomials over $\mathbb{R}$, would refute this form of SETH.

C Sparsity for −1/1 Probabilistic Polynomials

In many settings, the choice of basis (which field element corresponds to "true" and which to "false") can have a large impact on the sparsity of polynomial representations of Boolean functions. For instance, over $\{0,1\}$, computing XOR exactly on n inputs requires $\Omega(2^n)$ monomials, whereas over $\{-1,1\}$ it requires only one monomial.

We will show in this section that the choice of basis can make a big difference in the sparsity of probabilistic polynomials for the complements function (defined in Section 2). Although Theorem 16 holds over any choice of basis for our polynomials, our proof of Theorem 5 relies heavily on the fact that we are working over a basis where "false" corresponded to 0, and hence setting a variable to false eliminated every monomial that it was a part of.

Suppose we look instead at the $\{-1,1\}$ basis, with the standard convention that "true" corresponds to -1. In this setting, the statement of Theorem 5 is no longer true, since we can construct a probabilistic polynomial for the complements function with polynomial sparsity and polynomial error:

▶ **Lemma 28.** *For any $\varepsilon > 0$, and any commutative ring R which doesn't have characteristic 2, the AND of n variables has a probabilistic polynomial over R of sparsity $O(1/\varepsilon)$ and error ε over the basis $\{-1,1\}$.*

Proof. Pick a random subset $S \subseteq \{1, \ldots, n\}$, and consider the polynomial $E(x_1, \ldots, x_n) = 1 + (-1)^{|S|} \prod_{i \in S} x_i$. If $AND(x) = 1$, then some nonempty subset of the x_i's are 1's, and the probability that $\prod_{i \in S} x_i = (-1)^{|S|}$ is $1/2$. Hence, in this case, $E = 0$ with probability $1/2$. If $AND(x) = -1$, then all x_i's are -1, and we always have that $E = 2$. Hence, if we take $k = \lceil \log(1/\varepsilon) \rceil$ independent copies $E_1, \ldots, E_k$ of the above E, then the polynomial $P := 1 - \frac{1}{2^{k-1}} \prod_{i=1}^{k} E_i$ has the desired properties. ◀

▶ **Theorem 29.** *For any $\varepsilon > 0$, and any commutative ring R which doesn't have characteristic 2, $COMP$ has a probabilistic polynomial over R of sparsity $O(1/\varepsilon)$ and error ε over the basis $\{-1, 1\}$.*

Proof. The XOR of two variables is a single monomial over the $\{-1, 1\}$ basis, and so we simply substitute these monomials into the probabilistic polynomial from Lemma 28. ◀

Theorem 29 with $\varepsilon = 1/\text{poly}(n)$ shows that $COMP$ has a probabilistic polynomial with polynomial sparsity and any polynomial error over the $\{-1, 1\}$ basis, whereas Theorem 5 says that no such probabilistic polynomial can exist over the $\{0, 1\}$ basis.

Finally, we remark that just as in the $\{0, 1\}$ case, we can still construct probabilistic polynomials for OR over $\{-1, 1\}$ that achieve any two of the three goals of low sparsity, low degree, and low error:

- Low sparsity and error comes from Theorem 29,
- Low degree and error still follows from Theorem 2 with $t = (\log n)/(p-1)$, since degree and error are unchanged upon changing basis, and
- Low sparsity and degree still follows from Theorem 2 with $t = 1$, since then the degree is still $(p-1)$ and so by exploiting multilinearity, the sparsity must be $O(n^{p-1})$.

D Separation Between Probabilistic Rank and Probabilistic Sparsity

▶ **Definition 30.** *Let $a : [2^{n/2}] \to \{0,1\}^{n/2}$ be the canonical bijection, R be any ring, $f : \{0,1\}^n \to \{0,1\}$ be any Boolean function, and $A \cup B = \{x_1, \ldots, x_n\}$ be any partition of the input variables to f into two sets of size $|A| = |B| = n/2$. For $x, y \in \{0,1\}^{n/2}$, let $f(A|_x, B|_y)$ denote f evaluated at the assignment where the variables in A are set to x and the variables in B are set to y. The* truth table matrix *$M_{f,A,B}$ of f is the $2^{n/2} \times 2^{n/2}$ matrix given by $M_{f,A,B}(i,j) = f(A|_{a(i)}, B|_{b(j)})$.*

For $\varepsilon \geq 0$, and any $2^{n/2} \times 2^{n/2}$ matrix M over R, the ε-probabilistic rank of M is the minimum r for which there is a distribution $\mathcal{D}$ on $2^{n/2} \times 2^{n/2}$ matrices over R of rank at most r such that for all i, j:

$$\Pr_{D \sim \mathcal{D}}[D(i,j) \neq M(i,j)] \leq \varepsilon.$$

The ε-probabilistic rank *of f is the maximum, over all such partitions $A \cup B$, of the ε-probabilistic rank of $M_{f,A,B}$ over R.*

In this section, we compare the probabilistic sparsity of a function with its probabilistic rank. We know that the probabilistic sparsity is always an upper bound on the probabilistic rank:

▶ **Lemma 31** ([4] Corollary 2.1)**.** *For any ring R, if $f \in R$-$\mathbf{SE}[m, \varepsilon]$, then f has ε-probabilistic rank at most m over R.*

Using our probabilistic polynomial sparsity lower bounds, we are able to prove a separation in two different settings.

D.1 Complements

First, by using Theorem 5, we can give a simple explicit function, the *complements function* (defined in Section 2), which has substantially lower probabilistic rank than probabilistic sparsity over $\mathbb{F}_2$.

Recall from Theorem 5 that $COMP$ requires superpolynomial sparsity for any polynomial error over $\mathbb{F}_2$. In contrast, EQ is known to have low probabilistic rank, by simulating a communication complexity protocol for disjointness (similar to our proof of Theorem 29), and it follows that $COMP$ has low probabilistic rank as well:

▶ **Proposition 32** ([4] Lemma D.2)**.** *EQ has ε-probabilistic rank $O(1/\varepsilon)$ over any field, including $\mathbb{F}_2$.*

▶ **Proposition 33.** *$COMP$ has ε-probabilistic rank $O(1/\varepsilon)$ over any field, including $\mathbb{F}_2$.*

Proof. The communication matrix of $COMP$ is the same as that of EQ, up to a permutation of columns that swaps y with $\bar{y}$ for each $y \in \{0,1\}^{\log n}$. ◀

▶ **Corollary 34.** *$COMP$ has polynomial rank for any polynomial error over $\mathbb{F}_2$, whereas it requires superpolynomial sparsity for any polynomial error over $\mathbb{F}_2$.*

Proof. Follows from Corollary 33 and Theorem 5. ◀

D.2 Compositions with high fan-in XORs

Second, we combine Corollary 19 with Lemma 31 and a simple upper bound construction to prove a separation between probabilistic rank and probabilistic sparsity for functions of the form $f \circ XOR_k$, over $\mathbb{F}_2$.

Let $f : \{0,1\}^n \to \{0,1\}$ be any Boolean function, and $\varepsilon > 0$, $k \geq 1$ be any values. Let d be the smallest value such that f has an $\mathbb{F}_2$-probabilistic polynomial of error ε and degree d. We will prove a separation for the nk-variate function

$$f \circ XOR_k(x_{1,1}, \ldots, x_{n,k}) := f\left(\bigoplus_{j=1}^{k} x_{1,j}, \ldots, \bigoplus_{j=1}^{k} x_{n,j}\right).$$

▶ **Proposition 35.** *Over $\mathbb{F}_2$, the function $f \circ XOR_k$ has ε-probabilistic rank at most*

$$r := \sum_{i=0}^{2d} \binom{2n}{i}.$$

Proof. Suppose we have partitioned our nk variables into parts A, B, and for $1 \leq i \leq n$, let $A_i \subseteq A$ and $B_i \subseteq B$ be the corresponding partition of the variables $x_{i,1}, \ldots, x_{i,k}$. Define $y_i := \bigoplus_{x_{i,j} \in A_i} x_{i,j} \in \{0,1\}$ and $z_i := \bigoplus_{x_{i,j} \in B_i} x_{i,j} \in \{0,1\}$, and note that y_i and z_i are each functions of the variables in only one of the parts.

By assumption, f has an $\mathbb{F}_2$-probabilistic polynomial P of error ε and degree d. Suppose we draw a $p \in P$, and consider the expression

$$p(\chi(y_1, z_1), \chi(y_2, z_2), \ldots, \chi(y_n, z_n)), \tag{5}$$

where $\chi : \{0,1\}^2 \to \{0,1\}$ is the exact degree-2 polynomial for computing XOR on two variables. Expression (5) is therefore a probabilistic expression for $f \circ XOR_k$, and it is a degree $2d$ polynomial in the $2n$ variables $y_1, \ldots, y_n$ and $z_1, \ldots, z_n$, where each depends only on the variables in one side of the partition. The probabilistic rank is hence at most r by Lemma 31. ◀

Meanwhile, recall from Corollary 19 that $f \circ XOR_k$ requires $\mathbb{F}_2$-probabilistic sparsity $\varepsilon \cdot k^d$. Combining these two bounds can show a separation, for instance:

▶ **Theorem 36.** *For any*

$$k > \left(\frac{n \cdot e}{d}\right)^2,$$

the ε-probabilistic sparsity of $f \circ XOR_k$ is strictly greater than its ε-probabilistic rank. In particular, as k increases, the ε-probabilistic sparsity grows unboundedly while ε-probabilistic rank remains fixed.

Proof. With this bound on k, we have

$$k^d > \left(\frac{n \cdot e}{d'}\right)^{2d} \geq \sum_{i=0}^{2d} \binom{2n}{i},$$

as desired, where the second inequality is a standard bound on binomial coefficients. We can make the separation arbitrarily big by making k sufficiently large, since increasing k does not change the rank upper bound (the bound from Proposition 35 has no dependence on k), but increases the sparsity lower bound to as large as we would like. ◀

We can now apply known upper and lower bounds to get separations for other explicit functions. Consider, for instance, the majority function:

▶ **Theorem 37** ([29, 32])**.** *$\mathbb{F}_2$-probabilistic polynomials for MAJ on n bits with error ε require degree $\Omega(\sqrt{n \log(1/\varepsilon)})$.*

▶ **Theorem 38** ([3])**.** *There is an $\mathbb{F}_2$-probabilistic polynomial for MAJ on n bits with error ε and degree $O(\sqrt{n \log(1/\varepsilon)})$.*

Combining these with Theorem 36 yields:

▶ **Theorem 39.** *There is a constant $c > 0$ such that, for any*

$$k > \frac{c}{\log(1/\varepsilon)} \cdot n,$$

the ε-probabilistic sparsity of $MAJ \circ XOR$ over $\mathbb{F}_2$, where MAJ has fan-in n and each XOR has fan-in k, is strictly greater than its ε-probabilistic rank. In particular, as k increases, the ε-probabilistic sparsity grows unboundedly while ε-probabilistic rank remains fixed.

E Relationship between Degree, Sparsity, and Rank

Since the degree and sparsity of a probabilistic polynomial are related concepts, we are able to bound one using bounds on the other. Bounding the degree of a probabilistic polynomial will bound the sparsity of the polynomial, since there are only so many monomials of a given degree:

▶ **Proposition 40.** *For any decision problem $\mathcal{F}$, any $d, \varepsilon > 0$, and any prime p, if $\mathcal{F} \in \mathbb{F}_p$-$DE[d, \varepsilon]$, then $\mathcal{F} \in \mathbb{F}_p$-$SDE[m, d, \varepsilon]$, where*

$$m := \sum_{i=0}^{d} \binom{n}{i} \leq (1+n)^d.$$

Proof. Let P be a $\mathbb{F}_p$-DE$[d, \varepsilon]$ representation of $\mathcal{F}$. Since $x_i^2 = x_i$ whenever $x_i \in \{0, 1\}$, we may assume that any polynomial in the support of P is multilinear. Hence, each polynomial in the support of P is a multilinear polynomial of degree at most d, which has at most m monomials. P is therefore a $\mathbb{F}_p$-SDE$[m, d, \varepsilon]$ representation of $\mathcal{F}$. ◀

Perhaps more surprisingly, decision problems with low sparsity probabilistic polynomials must have low degree representations as well:

▶ **Proposition 41.** *For any decision problem $\mathcal{F}$, any $m, \varepsilon > 0$, and any prime p, if $\mathcal{F} \in \mathbb{F}_p$-SE$[m, \varepsilon]$, then $\mathcal{F} \in \mathbb{F}_p$-DE$[d, 2\varepsilon]$, where*

$$d := (p-1)\log_p(m/\varepsilon).$$

Proof. Let P be a $\mathbb{F}_p$-SE$[m, \varepsilon]$ representation of $\mathcal{F}$. We design a new probabilistic polynomial for $\mathcal{F}$, defined as follows:

1. Draw a polynomial $p(x_1, \ldots, x_n)$ from P. We can write p as a sum of monomials as $p(x_1, \ldots, x_n) = \sum_{i=1}^{m'} a_i \cdot m_i$, where $m' \leq m$, each $a_i \in \mathbb{F}_p$, and each m_i is a monomial.
2. Recall by Theorem 2 that AND $\in \mathbb{F}_p$-SDE$[O(n^{(p-1)t}), (p-1)t, 1/p^t]$ for all $t \geq 1$. Draw a polynomial q from the corresponding probabilistic polynomial on n variables with $t = \log_p(m/\varepsilon)$.
3. Each monomial m_i computes the AND of some set $x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}$ of variables. It can therefore be probabilistically computed as $q(x_{i,1}, \ldots, x_{i,n_i}, 1, 1, \ldots, 1)$, where there are $(n - n_i)$ '1's. We can thus output the polynomial

$$\sum_{i=1}^{m'} a_i \cdot q(x_{i,1}, \ldots, x_{i,n_i}, 1, 1, \ldots, 1).$$

This has degree $(p-1)t = d$, and by a union bound over the error of the replacement for each monomial, it has error at most $\varepsilon + m/p^t = 2\varepsilon$. ◀

At most a logarithmic factor is lost when converting between these two bounds; if $\mathcal{F}$ has probabilistic degree d then it can have probabilistic sparsity $\leq n^d$, and if it has probabilistic sparsity n^d then it can have probabilistic degree $\leq d \cdot (p-1) \cdot \log_p(n)$.

Proposition 41 allows us to prove probabilistic sparsity lower bound from probabilistic degree lower bounds, which are much more common in the literature. For instance:

▶ **Corollary 42.** *If $f : \{0,1\}^n \to \{0,1\}$ is any Boolean function which requires $\mathbb{F}_2$-probabilistic degree d for error ε, then f requires $\mathbb{F}_2$-probabilistic sparsity $\varepsilon \cdot 2^d$ for error ε.*

Proof. A lower sparsity representation would yield a probabilistic degree less than d using Proposition 41. ◀

Combining Proposition 40 with Lemma 31, we see that if f has a probabilistic polynomial of degree d, then it has probabilistic rank at most $\sum_{i=1}^{d} \binom{n}{i}$. Applying ideas from the recent breakthrough work of Croot, Lev, and Pach [13] on Roth's theorem over $\mathbb{Z}_4^n$, we can derive an improved bound on the rank from the degree:

▶ **Proposition 43.** *Let n be even. Let $f : \{0,1\}^n \to \{0,1\}$ be any Boolean function with a probabilistic polynomial of degree d and error ε. Then, the ε-probabilistic rank of f is at most $m := 2\sum_{i=0}^{\lfloor d/2 \rfloor} \binom{n/2}{i}$.*

We note that Dvir and Edelman [14] also used this observation when studying the rigidity of a certain family of matrices. Proposition 43 will follow immediately from Theorem 45 below, but first we need a definition.

▶ **Definition 44.** *Let R be and ring, n be an even integer, and $a : [2^{n/2}] \to \{0,1\}^{n/2}$ be any bijection between the two sets. For any function $f : \{0,1\}^n \to R$, the* Boolean rank of f over R *is the minimum r such that there are matrices A, B of dimensions $2^{n/2} \times r$ and $r \times 2^{n/2}$ with $(A \cdot B)[i,j] = f(a(i), a(j))$ for all i, j.*

▶ **Theorem 45.** *Let n be even. Let P be a multilinear polynomial in n variables with degree at most d over a ring R. Then, the Boolean rank of P over R is at most $m := 2\sum_{i=0}^{\lfloor d/2 \rfloor} \binom{n/2}{i}$.*

Proof. Think of P as being over two sets of variables, $\{x_1, \ldots, x_{n/2}\}$ and $\{y_1, \ldots, y_{n/2}\}$. We begin with some notation: For sets $I, J \subseteq [n/2]$, let x^I denote $\prod_{i \in T} x_i$, and let $c_{I,J}$ be the coefficient in P of the monomial $x^I y^J$. Let $I_1, \ldots, I_t$ be a list of all subsets of $[n/2]$ of cardinality at most $\lfloor d/2 \rfloor$. Given a monomial M and a variable assignment $A \in \{0,1\}^{n/2}$, we let $M|_A$ denote the evaluation of M on the assignment A.

We want to prepare two matrices A ($2^{n/2} \times r$) and B ($r \times 2^{n/2}$), and prove for all i, j that

$$(A \cdot B)[i,j] = P(a(i), a(j)).$$

For each $i \in [2^{n/2}]$, we make a $2t$-dimensional vector

$$A[i,:] := \left(x^{I_1}|_{a(i)}, \ldots, x^{I_t}|_{a(i)}, \sum_{J \subseteq [n/2], |J| \leq |I_1|} c_{J,I_1} x^J|_{a(i)}, \ldots, \sum_{J \subseteq [n/2], |J| \leq |I_t|} c_{J,I_t} x^J|_{a(i)} \right).$$

Correspondingly, for all $j \in [2^{n/2}]$ we make an $2t$-dimensional vector

$$B[:,j] := \left(\sum_{J \subseteq [n/2], |J| < |I_1|} c_{I_1,J} y^J|_{a(j)}, \ldots, \sum_{J \subseteq [n/2], |J| < |I_t|} c_{I_t,J} y^J|_{a(j)}, y^{I_1}|_{a(j)}, \ldots, y^{I_t}|_{a(j)} \right).$$

Since every d-set of $\{x_1, \ldots, x_{n/2}, y_1, \ldots, y_{n/2}\}$ contains either at most $\lfloor d/2 \rfloor$ x_i's or at most $\lfloor d/2 \rfloor$ y_i's, we observe as desired that

$$\langle A[i,:], B[:,j] \rangle = \sum_{I,J \in \{I_1, \ldots, I_t\}} a_{I,J} x^I|_{a(i)} y^J|_{a(j)} = P(a(i), a(j)). \qquad \blacktriangleleft$$

F Missing Proof

▶ **Lemma 46.** *There is a setting to α, β, and γ which satisfies equations* (1), (2), (3), *and* (4) *from the proof of Theorem 16.*

Proof. Recall that d, e are fixed, and we may set $c \geq 1$ to be arbitrarily large.

Setting

$$\beta := \sqrt{\frac{2\ln(2)e\gamma}{c}},$$

inequality (2) immediately holds. Setting $\gamma := \alpha + 2$, we have $\frac{\alpha+1}{\gamma} = \frac{\alpha+1}{\alpha+2} < 1$, so (3) holds. To satisfy (1), we need

$$\frac{\alpha^2(1+\alpha)}{3\ln(2)\gamma^2} = \frac{\alpha^2 + \alpha^3}{\ln(2)(3\alpha^2 + 12\alpha + 12)} \geq d + e.$$

Setting $\alpha > 1$ to be sufficiently large satisfies this inequality. Finally, we turn to (4). Substituting in our above solution for β, we have

$$\frac{(1-\beta)c}{\gamma} = \frac{c - \sqrt{2\ln(2)}e^{1/2}\gamma^{1/2}c^{1/2}}{\gamma}.$$

As γ is only a function of d and e which are fixed constants, and c can be made arbitrarily large, the above expression can be made larger than e. This completes the proof. ◀

G Generalization to a-b basis

In this section, we generalize Theorem 16 to hold for a general basis, where we can choose any two elements $a, b \in R$ from our commutative ring R to represent true and false. What we prove is:

▶ **Reminder of Theorem 3.** *For every commutative ring R, every pair of distinct $a, b \in R$, and all $d \geq 1$ and $e > 0$, there is a $c \geq 1$ such that for all sufficiently large m, the AND function on m variables is not in R-$\mathbf{SDE}_{a,b}[2^{dm/c}, m/c, 2^{-em/c}]$.*

We begin by recalling and proving the version of the Schwartz-Zippel Lemma, Lemma 13, which we use in the proof of Theorem 16.

▶ **Reminder of Lemma 13.** *Let p be a nonzero multilinear n-variate polynomial over any commutative ring R, of total degree at most d. Then $\Pr_{x\sim\{0,1\}^n}[p(x) \neq 0] \geq 1/2^d$.*

Proof. Let $d' \leq d$ be the total degree of p. Without loss of generality, we may assume p has the monomial $x_1x_2\cdots x_{d'}$ with a nonzero coefficient. Consider any $\{0,1\}$ assignment to the remaining $n - d'$ variables. The resulting polynomial p' still has the monomial $x_1x_2\cdots x_{d'}$ with the same coefficient, and so it is a nonzero multilinear polynomial of degree d' in d' variables. It remains to show that p' is nonzero on at least one of the $2^{d'}$ assignments to the remaining d' variables, which will imply as desired that $\Pr_{x\sim\{0,1\}^n}[p(x) \neq 0] \geq 1/2^d$.

Let m be a monomial in p' of lowest degree with a nonzero coefficient, let $c \neq 0$ be its coefficient, and let $S \subseteq \{x_1, \ldots, x_{d'}\}$ be the set of variables in m (it may be that m is the constant term, in which case $S = \emptyset$). Then, consider the assignment which sets the variables in S to 1, and the remaining variables to 0. Our polynomial p' must evaluate to c on this assignment, since monomial m will have all variables set to 1, and all other monomials will have at least one variable set to 0 by how we chose m. This is the desired assignment since $c \neq 0$. ◀

By carefully examining our proof of Theorem 16, we see that we only used the fact that we were working with the basis $\{0,1\}$ in three ways:

- We need that substituting a value in for one variable in a monomial decreases the degree of that monomial by at least one; this is true regardless of what basis we are working over.
- We need that the correct output when the AND is false is 0, and that the correct output when AND is true is any nonzero value; by subtracting b from our polynomial we can always assume this is the case, while changing the number of monomials by at most one and leaving the degree unchanged.
- We need the Schwartz-Zippel lemma (Lemma 13) to hold for our choice of basis.

If we could generalize Lemma 13 to hold for any basis $\{a, b\}$, then our same proof would work as stated to prove Theorem 3. Unfortunately this is impossible in general: If $a, b \in R$ are such that $a - b$ is a zero-divisor, then Lemma 13 does not hold when we use the $\{a, b\}$ basis. For example, working over $R = \mathbb{Z}/6\mathbb{Z}$, with $(a, b) = (0, 3)$, the single variable linear polynomial $p(x) = 2x$ is a nonzero polynomial, but is zero on all inputs from the basis.

Although it is possible to salvage this method, at least for most choices of $\{a, b\}$, we will instead proceed in a different way, which does not rely on how our original proof works, except for the fact (as remarked) that Theorem 3 holds when the *input basis* (what values we input to the polynomial for true or false) is $\{0, 1\}$, and the *output basis* (what values the polynomial outputs for true or false) is $\{0, b\}$ for any nonzero $b \in R$.

We begin with the input basis $\{0, b\}$ for any nonzero $b \in R$, where true maps to b but false still maps to 0.

▶ **Lemma 47.** *For every commutative ring R, every nonzero $a \in R$, and all $d \geq 1$ and $e > 0$, there is a $c \geq 1$ such that for all sufficiently large m, the AND function on m variables is not in R-**SDE**$_{0,b}[2^{dm/c}, m/c, 2^{-em/c}]$.*

Proof. For a given d and e, we make the same choice of $c \geq 1$ as in Theorem 16. Assume to the contrary that AND is in R-**SDE**$_{0,b}[2^{dm/c}, m/c, 2^{-em/c}]$, and let p be the corresponding probabilistic polynomial. We will convert p into a R-**SDE**$_{0,1}[2^{dm/c}, m/c, 2^{-em/c}]$ representation, which will contradict Theorem 16.

Draw a polynomial q from p, and consider any nonzero monomial $\mathfrak{m}$ of q, with degree $\mathfrak{d}$ and nonzero coefficient $\mathfrak{a}$. If any of the variables in $\mathfrak{m}$ is assigned to 0, then $\mathfrak{m}$ evaluates to 0, and otherwise it evaluates to $\mathfrak{a} \cdot b^{\mathfrak{d}}$.

Consider the new polynonial q', given by

$$q'(x) = \sum_{\mathfrak{a} \cdot \mathfrak{m} \in q} \mathfrak{a} \cdot b^{\mathfrak{d}} \cdot \mathfrak{m}.$$

In other words, we have multiplied each monomial $\mathfrak{m}$ from q by $b^{\mathfrak{d}}$. Now, for any $x \in \{0, 1\}^n$, let $x' \in \{0, b\}^n$ be given by $x'_i = b \cdot x_i$. Then we always have that $q'(x) = q(x')$. Since x corresponds to the same true/false assignment over the $\{0, 1\}$ basis as x' does over the $\{0, b\}$ basis, this means that the resulting distribution on q' is a probabilistic polynomial with input basis $\{0, 1\}$ and output basis $\{0, b\}$ for AND with the same sparsity, degree, and error as p, as desired. ◀

Finally, we generalize to all input bases $\{a, b\}$.

Proof of Theorem 3. For a given $d \geq 1$ and $e > 0$, let $c(d, e)$ be the choice of c which is made by Lemma 47. We will choose $c = c(d+1, e)$ here. Similar to before, our plan is to convert from a R-**SDE**$_{a,b}[2^{dm/c}, m/c, 2^{-em/c}]$ representation to a R-**SDE**$_{0,b-a}[2^{(d+1)m/c}, m/c, 2^{-em/c}]$ representation, and then to invoke Lemma 47.

Assume to the contrary that AND is in R-**SDE**$_{a,b}[2^{dm/c}, m/c, 2^{-em/c}]$, and let p be the corresponding probabilistic polynomial. Draw a polynomial q from p, and consider the new polynomial q' given by

$$q'(x_1, x_2, \ldots, x_n) = q(x_1 + a, x_2 + a, \ldots, x_n + a).$$

Our desired probabilistic polynomial over the $\{0, b - a\}$ input basis will be the resulting distribution on q'. The polynomial q' clearly has the same degree as q, and errs on the same true/false inputs as q.

Consider any nonzero monomial $\mathfrak{m}$ of q, with degree $\mathfrak{d}$ and nonzero coefficient $\mathfrak{a}$. Each variable in $\mathfrak{m}$ has been replaced by the sum of two terms, and so $\mathfrak{m}$ has been replaced by $2^{\mathfrak{d}}$ monomials when expanded out. Since p had degree at most m/c, we have that $2^{\mathfrak{d}} \leq 2^{m/c}$, and so since q had at most $2^{dm/c}$ monomials, we have that q' has at most $2^{dm/c} \cdot 2^{m/c} = 2^{(d+1)m/c}$ monomials. Our probabilistic polynomial is therefore a R-**SDE**$_{0,b-a}[2^{(d+1)m/c}, m/c, 2^{-em/c}]$ representation, as desired. ◀

We note finally that, although we have been limiting ourselves to the same input basis $\{a, b\}$ for all of the n variables, we could actually choose a different input basis for each variable, and then another output basis like usual, and the above proof still holds.

A Bicriterion Concentration Inequality and Prophet Inequalities for k-Fold Matroid Unions

Noga Alon
Department of Mathematics, Princeton University, NJ, USA
Schools of Mathematics and Computer Science, Tel Aviv University, Tel Aviv, Israel

Nick Gravin
Key Laboratory of Interdisciplinary Research of Computation and Economics,
Shanghai University of Finance and Economics, China

Tristan Pollner
Department of Management Science and Engineering, Stanford University, CA, USA

Aviad Rubinstein
Department of Computer Science, Stanford University, CA, USA

Hongao Wang
Department of Computer Science, Purdue University, West Lafayette, IN, USA

S. Matthew Weinberg
Department of Computer Science, Princeton University, NJ, USA

Qianfan Zhang
Department of Computer Science, Princeton University, NJ, USA

Abstract

We investigate prophet inequalities with competitive ratios approaching 1, seeking to generalize k-uniform matroids. We first show that large girth does *not* suffice: for all k, there exists a matroid of girth $\geq k$ and a prophet inequality instance on that matroid whose optimal competitive ratio is $\frac{1}{2}$. Next, we show k-fold matroid unions *do* suffice: we provide a prophet inequality with competitive ratio $1 - O(\sqrt{\frac{\log k}{k}})$ for any k-fold matroid union. Our prophet inequality follows from an online contention resolution scheme.

The key technical ingredient in our online contention resolution scheme is a novel bicriterion concentration inequality for arbitrary monotone 1-Lipschitz functions over independent items which may be of independent interest. Applied to our particular setting, our bicriterion concentration inequality yields "Chernoff-strength" concentration for a 1-Lipschitz function that is not (approximately) self-bounding.

2012 ACM Subject Classification Theory of computation → Online algorithms

Keywords and phrases Prophet Inequalities, Online Contention Resolution Schemes, Concentration Inequalities

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.4

Related Version *Full Version*: https://arxiv.org/abs/2411.11741

Funding *Noga Alon*: Research supported in part by NSF grant DMS-2154082.
Nick Gravin: Research is supported by National Key R & D Program of China (2023YFA1009500), NSFC grant 61932002, "the Fundamental Research Funds for the Central Universities in China".
Aviad Rubinstein: Research supported in part by NSF CCF-1954927, and a David and Lucile Packard Fellowship.
S. Matthew Weinberg: Research supported in part by NSF CCF-1955205. During Professor Weinberg's development of this paper, he participated as an expert witness on behalf of the State of Texas in ongoing litigation against Google (the "Google Litigation").
Qianfan Zhang: Research supported in part by NSF CCF-1955205.

Acknowledgements The authors are grateful to the anonymous reviewers for helpful feedback on the initial submission of this work.



© Noga Alon, Nick Gravin, Tristan Pollner, Aviad Rubinstein, Hongao Wang, S. Matthew Weinberg, and Qianfan Zhang;
licensed under Creative Commons License CC-BY 4.0



1 Introduction

Prophet inequalities are fundamental problems in optimal stopping theory, whose study dates back to seminal work of Krengel and Sucheston [17], and that have wide applications across Economics and Computer Science (e.g., [8, 11]). A prophet inequality instance contains a ground set E of elements, a family $\mathcal{F} \subseteq 2^E$ of feasible sets, and a collection of distributions $\{\mathcal{D}_e\}_{e\in E}$. For one element at a time, a random variable v_e is drawn from distribution $\mathcal{D}_e$ independently and revealed to a gambler, who immediately and irrevocably decides whether to accept or reject e. The gambler must at all times maintain the set of accepted elements $A \in \mathcal{F}$, and gets payoff $\sum_{e\in A} v_e$ at the end of the game. A prophet inequality is c-competitive if it guarantees $\mathbf{E}[\sum_{e\in A} v_e] \geq c \cdot \mathbf{E}[\max_{S\in\mathcal{F}} \sum_{e\in S} v_e]$.[1]

Krengel and Sucheston's seminal result establishes a $\frac{1}{2}$-competitive prophet inequality for any instance where $\mathcal{F}$ is a 1-uniform matroid (i.e. at most one element is feasible to accept), and moreover establish that no better guarantee is possible.[2] For k-uniform matroids, however, a significantly improved guarantee of $1-O(\frac{1}{\sqrt{k}})$ is possible [1, 14, 10]. This motivates the following question: for a given $\varepsilon > 0$, what conditions on $\mathcal{F}$ suffice for a $(1-\varepsilon)$-competitive prophet inequality?

Main Result I: Large Girth does not Suffice

A natural starting point to address this question is to first understand what makes k-uniform matroids "special" in the sense that the canonical hard instance cannot be embedded. One conjecture might be because k-uniform matroids have large *girth*: there are *no infeasible sets of size* $\leq k$. So, a natural first question to ask is whether $\mathcal{F}$ having large girth suffices in order to conclude that any instance over $\mathcal{F}$ admits a c-competitive prophet inequality. Our first main result establishes that large girth does *not* suffice.

▶ **Theorem 1.** *For all $k \geq 1$ and $\varepsilon > 0$, there exists a prophet inequality instance $(E, \mathcal{F}, \{\mathcal{D}\}_{e\in E})$ such that: (a) $(E, \mathcal{F})$ is a graphic matroid with girth k, and (b) $(E, \mathcal{F}, \{\mathcal{D}\}_{e\in E})$ does not admit a $(\frac{1}{2}+\varepsilon)$-competitive prophet inequality.*

Our construction leverages dense graphs of high girth (and a particular construction of [18]) in order to effectively embed multiple copies of the canonical hard 1-uniform instance. See Section 3 for further details.

Main Result II: k-fold Matroid Unions Suffice

Theorem 1 motivates richer generalizations of k-uniform matroids. We next consider *k-fold matroid unions*, observing that k-uniform matroids are the union of k 1-uniform matroids. Given a matroid $\mathcal{M} = (E, \mathcal{F})$ over ground set E with feasible sets $\mathcal{F}$, the *k-fold union of* $\mathcal{M}$ is a new matroid $\mathcal{M}^k$ with ground set E and feasible sets $\mathcal{F}^k := \{F_1 \cup F_2 \cup \cdots \cup F_k : F_1, F_2, \ldots, F_k \in \mathcal{F}\}$. That is, a set is feasible in $\mathcal{M}^k$ if it can be partitioned into k sets that are each feasible in $\mathcal{M}$.

---

[1] The expectation is taken with respect to the random variables $\{v_e\}_{e\in E}$, which in turn makes A a random variable.

[2] That is, there exist prophet inequality instances over 1-uniform matroids for which better than a $\frac{1}{2}$-competitive ratio is impossible. The hard instance is quite simple: $v_1 \sim \mathcal{D}_1$ is a point mass at 1, and $v_2 \sim \mathcal{D}_2$ is equal to $\frac{1}{\varepsilon}$ with probability ε and 0 otherwise. A gambler who sees v_1 first cannot achieve expect reward exceeding 1, but a prophet who always takes the maximum can achieve expected reward of $2-\varepsilon$.

▶ **Theorem 2.** *For every prophet inequality instance $(E, \mathcal{F}^k, \{\mathcal{D}_e\}_{e\in E})$ where $(E, \mathcal{F}^k)$ is the k-fold union of a matroid $(E, \mathcal{F})$, there exists a $(1 - O(\sqrt{\frac{\log k}{k}}))$-competitive prophet inequality.*

Our proof of Theorem 2 follows from a novel Online Contention Resolution Scheme (OCRS). An OCRS is parameterized by a ground set E, a feasibility family $\mathcal{F}$, and a vector of probabilities $\boldsymbol{x} \in \text{ConvexHull}(\{\mathbf{1}_F : F \in \mathcal{F}\}) \subseteq [0,1]^E$ (that is, $\boldsymbol{x}$ can be written as a convex combination of indicator vectors of feasible sets). One at a time, elements of E are revealed and *active* with probability x_e independently. If an element is active, it can be accepted or rejected (if inactive, it must be rejected), and the accepted elements must at all times be in $\mathcal{F}$. An OCRS is c-selectable if every element e is accepted with probability at least $c \cdot x_e$. In this language, Theorem 2 follows from a novel $(1 - O(\sqrt{\frac{\log k}{k}}))$-selectable OCRS for k-fold matroid unions.

To prove our OCRS, we follow a similar framework as [12], and design a recursive decomposition of $\mathcal{F}$ over which to greedily accept active elements. There are two key challenges to applying their framework, which we overview in greater detail in Subsection 4.2. We give a representative example below.

Applied to the 1-uniform matroid, the [12] algorithm simply proposes "accept any active element independently with probability b." Then, linearity of expectation suffices to observe that there are at most b elements in expectation that are both active and accepted,[3] and Markov's inequality suffices to guarantee that with probability at least $1 - b$, no elements are accepted at all. This suffices to guarantee that for all e: (a) with probability at least $1 - b$ it is feasible to accept e when revealed, and (b) independently, we will accept e with probability b conditioned on e being active and feasible. This implies a $b(1 - b)$-selectable algorithm, which is optimized at $b = \frac{1}{2}$.

Applied to the k-uniform matroid, a natural algorithm would again be "accept any active element independently with probability b." Then, linearity of expectation still suffices to observe that there are at most bk elements in expectation that are both active and accepted, but Markov's inequality only guarantees that with probability at least $1-b$, at most k elements are accepted. This would lead to the same $\frac{1}{4}$-selectable OCRS, which is not the desired $1 - O(\sqrt{\frac{\log k}{k}})$. Of course, the obvious fix is to use a significantly stronger concentration inequality than Markov's. E.g., a Chernoff bound suffices to guarantee that with probability at least $1 - \frac{1}{k}$ at most $k - 1$ elements are accepted, when $b = 1 - O(\sqrt{\frac{\log k}{k}})$. This leads to the desired $1 - O(\sqrt{\frac{\log k}{k}})$ selectable OCRS for *k-uniform* matroids. However, Chernoff bounds are insufficient for the general class of k-fold matroid unions – the probability that a particular element is feasible to accept is a highly combinatorial function that depends on the underlying matroid structure. Thus our Theorem 2 has two components: first, a decomposition that reduces the OCRS problem to a concentration inequality and second, a novel concentration inequality, which is our third main result.

Main Result III: A Bicriterion Concentration Inequality

Putting aside prophet inequalities for a moment, concentration inequalities are a core aspect of applied probability with widespread application across many areas of Computer Science. One representative setting is the following: Let $f : \{0,1\}^E \to \mathbb{R}$ be some function, and let $\boldsymbol{X} = \langle X_e \rangle_{e\in E}$ be a vector of independent Bernoulli random variables, where $X_e \sim \text{Ber}(p_e)$. A canonical question asks: what is the probability that $f(\boldsymbol{X})$ exceeds $\mathbf{E}[f(\boldsymbol{X})] + t$?

[3] There are at most 1 elements in expectation that are active, and each active element is accepted with probability b.

On one extreme, McDiarmid's inequality holds whenever f is 1-Lipschitz. On the other, Chernoff bounds are *significantly* stronger, if f is linear (and 1-Lipschitz). In between, "Chernoff-strength" concentration holds whenever f is fractionally-subadditive or (approximately) self-bounding [5, 6, 22, 7, 27], but this provably does not extend even to the case when f is subadditive [27].

Our third main result provides a *bicriterion concentration inequality* for any monotone 1-Lipschitz function. Specifically, if $\boldsymbol{X}$ is a vector of Bernoulli random variables with probability vector $\boldsymbol{p}$, let $\boldsymbol{X}^{(s)}$ denote a vector of Bernoulli random variables with probability vector $e^{-s}\boldsymbol{p}$. That is, each probability p_i has been decreased by a factor of e^{-s}. Our new concentration inequality establishes:

▶ **Theorem 3.** *Let $f : \{0,1\}^E \to \mathbb{R}$ be a monotone 1-Lipschitz function. For any $s \in (0,1]$, $t > 0$:*

$$\mathbf{Pr}\left[f(\boldsymbol{X}^{(s)}) \geq \mathbf{E}[f(\boldsymbol{X})] + t\right] \leq e^{-st}.$$

A helpful comparison point is McDiarmid's inequality, which instead proves the following: $\mathbf{Pr}\left[f(\boldsymbol{X}) \geq \mathbf{E}[f(\boldsymbol{X})] + t\right] \leq e^{-2t^2/|E|}$. The distinctions are: (a) our concentration inequality is bicriterion – we analyze $f(\boldsymbol{X}^{(s)})$ instead of $f(\boldsymbol{X})$, and (b) our concentration has an exponent of $-st$ instead of $-2t^2/|E|$. In particular, McDiarmid's inequality depends on the dimension $|E|$ and cannot possibly kick in for $t \ll \sqrt{|E|}$, whereas our concentration inequality can kick in for any $t > 1/s$. A representative example to have in mind might be $s = \sqrt{\log(1/\varepsilon)/\,\mathbf{E}[f(\boldsymbol{X})]}$ and $t = \sqrt{\log(1/\varepsilon)\,\mathbf{E}[f(\boldsymbol{X})]}$. This results in a tail probability of ε for exceeding $\mathbf{E}[f(\boldsymbol{X})]$ by $\sqrt{\log(1/\varepsilon)}$ multiples of $\sqrt{\mathbf{E}[f(\boldsymbol{X})]}$, which is "Chernoff-strength". *But*, this concentration holds only for $f(\boldsymbol{X}^{(s)})$, rather than $f(\boldsymbol{X})$. This suffices for our application.

To prove Theorem 3, we utilize the entropy method for self-bounding functions [5, 6, 22, 7] in an unconventional way. We give a more detailed technical overview in Subsection 5.1.

1.1 Related Work

There are three strands of related work: prophet inequalities, concentration inequalities, and attempts to generalize k-uniform guarantees.

Prophet Inequalities

Prophet inequalities have a long history in Mathematics, Computer Science, and Operations Research. Representative results include Krengel and Sucheston's initial $\frac{1}{2}$-approximation [17], Samuel-Cahn's elegant thresholding strategy [25], Chawla et al.'s connection to Bayesian mechanism design [8], Kleinberg and Weinberg's extension to matroids [16], and Dutting et al.'s connection to Price of Anarchy [11].

Of particular relevance to our work are prophet inequalities for k-uniform matroids. The first $1 - O(\sqrt{\frac{\log k}{k}})$ approximation was developed by [13], and the first asymptotically tight $1 - O(\frac{1}{\sqrt{k}})$ approximation was developed by [1]. Subsequent works achieve the same $1 - O(\frac{1}{\sqrt{k}})$ approximation with sample access [4], the optimal OCRS [14], or a simpler OCRS [10]. The most technically related paper to our work is [12], whose OCRS framework we leverage. It remains an open question whether the prophet inequality for k-fold matroid unions can be improved to $1 - O(\frac{1}{\sqrt{k}})$.

Concentration Inequalities

The most related concentration inequalities fit the same framework but consider different f. McDiarmid's inequality [21] holds for all 1-Lipschitz f, Schechtman's inequality holds for f that are subadditive [26], Bucheron et al. derive an inequality for f that are self-bounding functions [5], and Vondrák derives an inequality for f that are fractionally subadditive [27]. These inequalities are commonly used across Theoretical Computer Science, and especially within combinatorial prophet inequalities and Bayesian mechanism design [24, 23].

Generalizing k-uniform matroids

Recent work of [9] considers (offline) contention resolution and correlation gap inequalities. Here too, guarantees for k-uniform matroids are significantly stronger than what is achievable for arbitrary matroids. Their work similarly extends guarantees achievable for k-uniform matroids to k-fold matroid unions. In comparison to our work: (a) the general motivation is the same – both works seek to extend stronger guarantees for k-uniform matroids to more general settings, (b) the problems studied and technical aspects are orthogonal,[4] (c) our work also proposes a bicriterion concentration inequality.

Another generalization of k-uniform matroids are *packing constraints*, where each element has a d-dimensional size in $[0,1]^d$ and one can accept a subset of elements if their size vectors sum to at most k in every coordinate. Packing constraints have been studied in various online settings, including secretary model [15], prophet model [2], and mixed model [3].

2 Preliminaries

Prophet Inequalities

In the *prophet inequality* problem, we are given a ground set of elements E, a downward-closed family of feasible sets $\mathcal{F} \subseteq 2^E$, and a distribution $\mathcal{D}_e$ associated with each element $e \in E$. Elements arrive in an adversarial order.[5] As each element e arrives, its value v_e, independently drawn from $\mathcal{D}_e$, is revealed. At this point, an irrevocable decision must be made whether to include e in its output A, while keeping $A \in \mathcal{F}$.

For $c \in [0,1]$, we say an online algorithm implies a *c-competitive* prophet inequality for $\mathcal{F}$, if for any distributions $\{\mathcal{D}_e\}_{e\in E}$,

$$\mathbf{E}\left[\sum_{e\in A} v_e\right] \geq c \cdot \mathbf{E}\left[\max_{S\in\mathcal{F}} \sum_{e\in S} v_e\right]$$

where the expectation is taken with respect to random variables $\{v_e\}_{e\in E}$ and the internal randomness of the algorithm.

Online Contention Resolution Schemes

Given a ground set of elements E and a downward-closed family of feasible sets $\mathcal{F} \subseteq 2^E$, we define the polytope of $\mathcal{F}$ as the convex hull of all characteristic vectors of feasible sets, i.e., $\mathcal{P}_\mathcal{F} = \text{ConvexHull}(\{\mathbf{1}_F : F \in \mathcal{F}\}) \subseteq [0,1]^E$.

[4] While in principle, contention resolution and online contention resolution may appear similar, the relevant techniques are fundamentally different with little overlap. Similarly, while correlation gap inequalities are sometimes a useful tool in prophet inequalities, in this case there is no overlap.

[5] There are various adversarial models. The weakest is the *fixed-order adversary*, which sets the arrival order offline, based solely on the distributions. The strongest is the *almighty adversary*, which sets the arrival order online, with full knowledge of all realizations of randomness and the algorithm's past decisions. Our negative result in Section 3 applies to fixed-order adversary, while our positive result in Section 4 holds against almighty adversary.

An *online contention resolution scheme* (OCRS) takes a vector $\boldsymbol{x} \in \mathcal{P}_{\mathcal{F}}$ as input. Let $R(\boldsymbol{x}) \subseteq E$ be a random set where each element $e \in E$ is in $R(\boldsymbol{x})$ independently with probability x_e. The OCRS sees membership in $R(\boldsymbol{x})$ of elements in E, arriving in an adversarial order; when each $e \in E$ arrives, if $e \in R(\boldsymbol{x})$ (i.e., e is "active"), the scheme must decide irrevocably whether to include e in its output A, while keeping $A \in \mathcal{F}$.

For $c \in [0, 1]$, an OCRS is called *c-selectable* for $\mathcal{F}$, if for any $\boldsymbol{x} \in \mathcal{P}_{\mathcal{F}}$,

$$\mathbf{Pr}[e \in A \mid e \in R(\boldsymbol{x})] \geq c \quad \forall e \in E$$

where $A \in \mathcal{F}$ is the output of the OCRS, and the probability is measured with respect to $R(\boldsymbol{x})$ and internal randomness of the OCRS. As shown in [12], a c-selectable OCRS directly implies a c-competitive prophet inequality.

▶ **Lemma 4** ([12]). *For a ground set E and a family of feasible sets $\mathcal{F} \subseteq 2^E$, a c-selectable OCRS for $\mathcal{F}$ implies a c-competitive prophet inequality for $\mathcal{F}$.*

Matroids

A *matroid* $\mathcal{M} = (E, \mathcal{I})$ is defined by a ground set of elements E and a non-empty downward-closed family of independent sets $\mathcal{I} \subseteq 2^E$ with the *exchange property*, i.e., for every $A, B \in \mathcal{I}$ where $|A| > |B|$, there exists an element $e \in A \setminus B$ such that $B \cup \{e\} \in \mathcal{I}$. Given a matroid $\mathcal{M} = (E, \mathcal{I})$, the following notations are used throughout the paper:

- The *rank* of a set $S \subseteq E$ is the size of the largest independent set contained in S: $\text{rank}(S) = \max\{|I| : I \subseteq S, I \in \mathcal{I}\}$.
- The *span* of a set $S \subseteq E$ is the set of elements that is not independent from S: $\text{span}(S) = \{e \in E : \text{rank}(S) = \text{rank}(S \cup \{e\})\}$.
- The *restriction* of $\mathcal{M}$ to a set $S \subseteq E$ is a matroid $\mathcal{M}|_S = (S, \mathcal{I}|_S) = (S, \{I \in \mathcal{I} : I \subseteq S\})$.
- The *girth* of $\mathcal{M}$ is the size of the smallest dependent set: $\text{girth}(\mathcal{M}) = \min\{|S| : S \subseteq E, S \notin \mathcal{I}\}$.

Following are some special matroids that we will use later.

▶ **Example 5** (Uniform matroid). A *k-uniform matroid* $\mathcal{M} = (E, \mathcal{I})$ is a matroid in which the independent sets are exactly the sets that contains at most k elements for an integer $k \geq 1$, i.e, $\mathcal{I} = \{I \subseteq E : |I| \leq k\}$.

▶ **Example 6** (Graphical matroid). A *graphical matroid* $\mathcal{M} = (E, \mathcal{I})$ is a matroid in which the independent sets are the forests in a given undirected graph $G = (V, E)$, i.e., $\mathcal{I} = \{I \subseteq E : I \text{ is acyclic in } G\}$.

We formally define k-fold matroid union as follows.

▶ **Definition 7** (k-fold matroid union). *Given a matroid $\mathcal{M} = (E, \mathcal{I})$ and an integer $k \geq 1$, the* k-fold union *of $\mathcal{M}$ is defined as $\mathcal{M}^k = \underbrace{\mathcal{M} \vee \mathcal{M} \vee \cdots \vee \mathcal{M}}_{k \text{ times}} = (E, \mathcal{I}^k)$ where*

$$\mathcal{I}^k = \{I_1 \cup I_2 \cup \cdots \cup I_k : I_1, I_2, \ldots, I_k \in \mathcal{I}\}.$$

In other words, a set I is independent in $\mathcal{M}^k$ if and only if I can be partitioned into at most k independent sets in $\mathcal{M}$. Note that $\mathcal{M}^k$ remains a matroid by the closure property of matroid union.

3 Large Girth is Not Sufficient

In this section, we prove that a large girth is not sufficient for matroids (specifically, graphical matroids) to have a prophet inequality with a competitive ratio better than $\frac{1}{2}$.

▶ **Theorem 1.** *For all $k \geq 1$ and $\varepsilon > 0$, there exists a prophet inequality instance $(E, \mathcal{F}, \{\mathcal{D}\}_{e\in E})$ such that: (a) $(E, \mathcal{F})$ is a graphic matroid with girth k, and (b) $(E, \mathcal{F}, \{\mathcal{D}\}_{e\in E})$ does not admit a $(\frac{1}{2} + \varepsilon)$-competitive prophet inequality.*

To construct a hard instance, we start with a dense graph of large girth. We then transform the graph by splitting each edge (v_1, v_2) into two edges (v_1, u) and (v_2, u), where u is a newly introduced vertex. We obtain the final hard instance of the prophet inequality problem by embedding the hard instance of the single-item case into each of these edge pairs $(v_1, u), (v_2, u)$.

The hardness of this instance arises from the following observation: without accepting both edges in a pair, the instance essentially reduces to $|E|$ independent hard instances of the single-item case. On the other hand, one can accept at most $|V| - 1$ extra pairs of edges (in addition to $|E|$ single-item problems) at the same time without forming a cycle, which could not contribute a lot to the final solution because the graph is dense.

Proof of Theorem 1. We employ a construction of [18] which provides dense graphs of large girth. In particular, we will use that for any fixed k there exists some arbitrarily large n such that there is a graph G_n on n vertices with at least $n \log n$ edges and girth at least k.[6] Specifically, consider the graph G_n with vertices $V(G_n) = \{v_1, v_2, \ldots, v_n\}$ and edges $E(G_n) = \{e_1, e_2, \ldots, e_m\}$, where $m \geq n \log n$ and each edge $e_i = (a(i), b(i)) \in E(G_n)$ connects vertices $a(i)$ and $b(i)$ in $V(G_n)$. We construct a new graph H_n with $n + m$ vertices as follows:

- Begin with a set of $n+m$ vertices, labeled $V(H_n) := \{w_1, w_2, \ldots, w_n\} \sqcup \{u_1, u_2, \ldots, u_m\}$.
- For each edge e_i in G_n connecting $v_{a(i)}$ and $v_{b(i)}$, add in H_n an edge between u_i and $w_{a(i)}$ (call it f_i) as well as an edge between u_i and $w_{b(i)}$ (call it f'_i).

Hence H_n has a total of $2m$ edges. For $1 \leq i \leq m$, let the associated random variable X_{f_i} of f_i be a constant 1, and let the associated random variable $X_{f'_i}$ of f'_i follow a distribution which takes a value of $\frac{1}{\varepsilon}$ with probability ε, and a value of 0 with probability $1 - \varepsilon$. We consider an instance of the prophet inequality problem where the online algorithm is presented edges in the order $(f_1, f'_1, f_2, f'_2, \ldots, f_m, f'_m)$.

We first lower bound $\mathrm{OPT}(H_n)$, the expected value the optimal offline algorithm gets on this instance. Note that an offline algorithm could simply look at each pair $\{f_i, f'_i\}$ and take whichever edge has higher realized weight; this cannot create a cycle because every edge selected will be incident to a vertex of degree 1. We hence have the bound

$$\mathrm{OPT}(H_n) \geq \sum_{i=1}^{m} \left(\varepsilon \cdot \frac{1}{\varepsilon} + (1 - \varepsilon) \cdot 1 \right) = m(2 - \varepsilon).$$

Fix an online algorithm $\mathcal{A}$, and we now give an upper bound on its expected performance $\mathcal{A}(H_n)$ on the instance. The lower bound relies on the following observation.

▷ Claim 8. There are at most $n - 1$ values of i in $\{1, 2, \ldots, m\}$ such that $\mathcal{A}$ accepts both f_i and f'_i.

[6] In fact, [18] prove a significantly stronger result, but the weaker version stated above suffices for our purposes.

Proof. Suppose there are at least n such values of i; call them i_1, i_2, …, i_n. As the original graph G has n vertices, and a forest on n vertices has at most $n-1$ edges, we clearly see that there is a cycle among $\{e_{i_1}, e_{i_2}, \ldots, e_{i_n}\}$. That however would imply there is a cycle in H; namely, follow the cycle that existed in G, but replace each edge e_i with the edge f_i followed by the edge f'_i. ◁

For each $1 \le i \le m$, we now consider cases for what $\mathcal{A}$ gets in expectation from $\{f_i, f'_i\}$ right after f_i arrives:

- If $\mathcal{A}$ rejects f_i, then it clearly gets in expectation at most 1 from $\{f_i, f'_i\}$ because $\mathbb{E}[X_{f'_i}] = 1$.
- If $\mathcal{A}$ accepts f_i and rejects f'_i, then it clearly gets weight at most 1 from $\{f_i, f'_i\}$.
- If $\mathcal{A}$ accepts f_i and accepts f'_i, then it clearly gets weight at most $1 + \frac{1}{\varepsilon}$ from $\{f_i, f'_i\}$.

Let C_1 denote the set of all $i \in [m]$ such that $\mathcal{A}$ rejects f_i, let C_2 denote the set of all $i \in [m]$ such that $\mathcal{A}$ accepts f_i and rejects f'_i, and let C_3 denote the set of all $i \in [m]$ such that $\mathcal{A}$ accepts f_i and f'_i. Note C_1, C_2, and C_3 are random (disjoint) sets that may depend on the values realized by $\{X_{f_i}, X_{f'_i}\}_{i=1}^m$ and any randomness in $\mathcal{A}$. By the above cases, we can see that in expectation, $\mathcal{A}$ gets score at most

$$\sum_{i \in C_1} 1 + \sum_{i \in C_2} 1 + \sum_{i \in C_3} \left(1 + \frac{1}{\varepsilon}\right) = |C_1| + |C_2| + |C_3| \cdot \left(1 + \frac{1}{\varepsilon}\right).$$

Although $|C_1|$, $|C_2|$, and $|C_3|$ are random variables, $|C_1| + |C_2| \le m$ always, and by Claim 8 we have $|C_3| \le n - 1$ always. Hence, in expectation (averaging over all possible realizations of C_1, C_2, and C_3), we can bound the performance of $\mathcal{A}$ on H_n by $\mathcal{A}(H_n) \le m + n\left(1 + \frac{1}{\varepsilon}\right)$. As n grows, we can compute

$$\liminf_{n \to \infty} \frac{\mathcal{A}(H_n)}{\mathrm{OPT}(H_n)} \le \lim_{n \to \infty} \frac{m + n\left(1 + \frac{1}{\varepsilon}\right)}{m(2 - \varepsilon)} = \frac{1}{2 - \varepsilon}.$$

Taking $\varepsilon \to 0$ demonstrates the claimed result. ◀

4 k-Fold Unions are Sufficient

Our main goal in the section is to construct a good OCRS for k-fold matroid unions (Theorem 9). Combining with the reduction from prophet inequalities to OCRSs by [12] (Lemma 4), this immediately implies the existence of good prophet inequality for all k-fold matroid unions (Theorem 2).

▶ **Theorem 9.** *There exists a $(1 - O(\sqrt{\frac{\log k}{k}}))$-selectable OCRS for any k-fold matroid union $\mathcal{M}^k$.*

Our OCRS for k-fold matroid unions builds on the chain decomposition approach used in the matroid OCRS by [12], outlined in Subsection 4.1. We overview our approach and highlight main difficulties in Subsection 4.2. The construction is then formally given and analyzed in Subsection 4.3, where the bicriterion concentration inequality in Section 5 is used to bound its selectability.

4.1 Recap: OCRS for general matroids

We briefly describe the idea of the $\frac{1}{4}$-selectable matroid OCRS by [12]. Specifically, they show that for any parameter $b \in (0,1)$, there exists a $(1-b)$-selectable OCRS for any matroid $\mathcal{M} = (E, \mathcal{I})$ and $\boldsymbol{x} \in b \cdot \mathcal{P}_{\mathcal{M}}$. Note that one can "scale down" a vector $\boldsymbol{x}$ from $\mathcal{P}_{\mathcal{M}}$ to $b \cdot \mathcal{P}_{\mathcal{M}}$ by only considering each element independently with probability b. Formally:

▶ **Fact 10.** *For $b, c \in (0,1)$ and any matroid $\mathcal{M}$, a c-selectable OCRS for all $\boldsymbol{x} \in b \cdot \mathcal{P}_{\mathcal{M}}$ implies a bc-selectable OCRS for all $\boldsymbol{x} \in \mathcal{P}_{\mathcal{M}}$.*

Therefore, it follows that a $b(1-b)$-selectable ORCS exists for any matroid $\mathcal{M}$ and $\boldsymbol{x} \in \mathcal{P}_{\mathcal{M}}$. By letting $b = \frac{1}{2}$, they obtain a $\frac{1}{4}$-selectable matroid OCRS.

The greedy algorithm

Let us start with the simple greedy algorithm that always accepts the active element whenever possible. When $\mathcal{M}$ is a 1-uniform matroid, the greedy algorithm is actually $(1-b)$-selectable for $\boldsymbol{x} \in b \cdot \mathcal{P}_{\mathcal{M}}$ (i.e., $\sum_{e \in E} x_e \le b$ since $\mathcal{M}$ is 1-uniform), since the selectability of an element $e \in E$ can be easily lower bounded as

$$\begin{aligned}\mathbf{Pr}[e \text{ is accepted} \mid e \text{ is active}] &\ge \mathbf{Pr}[\text{no other element is active} \mid e \text{ is active}] \\ &\ge \mathbf{Pr}[\text{no element is active}].\end{aligned}$$

The first inequality holds because when there is no active elements besides e, the greedy algorithm can always accept e even if it arrives at the end. The second inequality holds due to the independence between elements. Moreover, by Markov's inequality,

$$\begin{aligned}\mathbf{Pr}[\text{no element is active}] &= 1 - \mathbf{Pr}[|R(\boldsymbol{x})| \ge 1] \\ &\ge 1 - \mathbf{E}[|R(\boldsymbol{x})|] = 1 - \sum_{e \in E} x_e \ge 1 - b.\end{aligned}$$

(Recall that $R(\boldsymbol{x})$ is the set of active elements.)

The first half of argument applies when $\mathcal{M}$ is a general matroid: for every element $e \in E$,

$$\mathbf{Pr}[e \text{ is accepted} \mid e \text{ is active}] \ge \mathbf{Pr}[e \notin \operatorname{span}(R(\boldsymbol{x}))].$$

However, unlike in 1-uniform matroids, the probability that an element $e \in E$ is spanned by active elements $R(\boldsymbol{x})$ could be much smaller than $1-b$, even for a scaled $\boldsymbol{x} \in b \cdot \mathcal{P}_{\mathcal{M}}$. In fact, the selectability of the greedy algorithm can be arbitrarily bad for a general matroid $\mathcal{M}$ (see, e.g., [19]).

Protection

Consider Algorithm 1, a modified greedy algorithm with a *protection set* $S \subsetneq E$ that only handles elements in $E \setminus S$. Intuitively, the algorithm accepts every active element $e \in E \setminus S$ whenever it does not conflict with any element in S. As a result, elements in S are "prioritized" over those in $E \setminus S$: regardless of which independent set from S is accepted, it remains an independent set when combined with the accepted elements in $E \setminus S$.

Algorithm 1 Modified greedy algorithm for $\mathcal{M} = (E, \mathcal{I})$ with a protection set $S \subseteq E$.

$A \leftarrow \emptyset$ ▷ *the set of accepted elements in* $E \setminus S$
for each arriving active element $e \in E \setminus S$ **do**
  if $e \notin \text{span}(A \cup S)$ **then**
    $A \leftarrow A \cup \{e\}$ ▷ *accepts element* e

For the modified greedy algorithm, we can similarly lower bound the selectability for $e \in E \setminus S$:

$$\mathbf{Pr}[e \text{ is accepted} \mid e \text{ is active}] \geq \mathbf{Pr}[e \notin \text{span}(R(\boldsymbol{x}) \cup S)].$$

The good news is that, such probabilities can be further lower bounded by $1 - b$ for the S obtained using Algorithm 2, an iterative algorithm that updates S by adding an element e whenever $\mathbf{Pr}[e \in \text{span}(R(\boldsymbol{x}) \cup S)] > b$.

Algorithm 2 Find a protection set S for $\mathcal{M} = (E, \mathcal{I})$ and $\boldsymbol{x} \in b \cdot \mathcal{P}_{\mathcal{M}}$.

function PROTECT($\mathcal{M}, \boldsymbol{x}, b$)
  $S \leftarrow \emptyset$
  while $\exists e \in E \setminus S,\ \mathbf{Pr}[e \in \text{span}(R(\boldsymbol{x}) \cup S)] > b$ **do**
    $S \leftarrow S \cup \{e\}$
  return S

Note that Algorithm 2 always terminates since E is a finite set, and the modified greedy algorithm with this protection set S guarantees $(1-b)$-selectability for every element $e \in E \setminus S$. More importantly, the protection is non-trivial, i.e., S is a proper subset of E.

▶ **Lemma 11** ([12]). *For any matroid* $\mathcal{M} = (E, \mathcal{I})$ *and* $\boldsymbol{x} \in b \cdot \mathcal{P}_{\mathcal{M}}$, PROTECT$(\mathcal{M}, \boldsymbol{x}, b) \subsetneq E$.

Therefore, it remains to get a good OCRS for $\mathcal{M}|_S$ and $\boldsymbol{x}|_S$, the restriction of the original matroid and vector to the protection set S.

Chain decomposition

The matroid OCRS in [12] starts with an offline prepossessing that finds the following *chain decomposition* of the elements:

$$\emptyset = N_\ell \subsetneq N_{\ell-1} \subsetneq \cdots \subsetneq N_1 \subsetneq N_0 = E$$

where $N_{i+1} = \text{PROTECT}(\mathcal{M}|_{N_i}, \boldsymbol{x}|_{N_i}, b)$ for every $0 \leq i < \ell$. And the OCRS is then operates by invoking Algorithm 1 on matroid $\mathcal{M}|_{N_i}$ with a protection set N_{i+1} for each $e \in N_i \setminus N_{i+1}$.

It is easy to see that these algorithms together produces an independent set of $\mathcal{M}$, and the selectability for each element $e \in N_i \setminus N_{i+1}$ is

$$\mathbf{Pr}[e \text{ is accepted} \mid e \text{ is active}] \geq 1 - \mathbf{Pr}[e \in \text{span}_{\mathcal{M}|_{N_i}}(R(\boldsymbol{x}|_{N_i}) \cup N_{i+1})] \geq 1 - b$$

where the last inequality holds due to the way N_{i+1} is obtained using Algorithm 2. By setting $b = \frac{1}{2}$, the resulting OCRS is $\frac{1}{2}$-selectable given any matroid $\mathcal{M}$ and $\boldsymbol{x} \in \frac{1}{2} \cdot \mathcal{P}_{\mathcal{M}}$.

4.2 Overview of our construction

We now give a high-level overview of our construction and highlight main difficulties. Let us first examine the case when $\mathcal{M}$ is a k-uniform matroid and see why the simple greedy algorithm works better for larger k.

Intuition from k-uniform matroids

When $\mathcal{M}$ is a k-uniform matroid, it turns out that the simple greedy algorithm that always accepts the active element whenever possible yields an OCRS with a selectability of $1 - O(\sqrt{\frac{\log k}{k}})$. To see this, consider the following tighter analysis of selectability for k-uniform matroids: for every element $e \in E$,

$$\mathbf{Pr}[e \text{ is accepted} \mid e \text{ is active}] \geq \mathbf{Pr}[e \notin \text{span}(R(\boldsymbol{x}))] = \mathbf{Pr}[|R(\boldsymbol{x})| < k].$$

Intuitively, $|R(\boldsymbol{x})|$ represents the number of slots occupied by active elements, and we know $|R(\boldsymbol{x})| < k$ indicates $e \notin \text{span}(R(\boldsymbol{x}))$. We want the bad event $|R(\boldsymbol{x})| \geq k$ to occur with a small probability.

Note that $|R(\boldsymbol{x})|$ is a sum of Bernoulli random variables and it concentrates very well: if we consider a slightly scaled-down $\boldsymbol{x} \in (1 - O(\sqrt{\frac{\log k}{k}})) \cdot \mathcal{P}_{\mathcal{M}}$, Chernoff bound (Theorem 27) tells us that $\mathbf{Pr}[|R(\boldsymbol{x})| \geq k] \leq \frac{1}{k}$. By Fact 10, one can further derive an OCRS for k-uniform matroids with a selectability of $(1 - O(\sqrt{\frac{\log k}{k}}))(1 - \frac{1}{k}) = 1 - O(\sqrt{\frac{\log k}{k}})$.

To summarize, the greedy algorithm performs well on k-uniform matroids because of the existence of a **fine-grained occupancy indicator $|R(\boldsymbol{x})|$ that concentrates well**.

Main idea and challenges

For a k-fold matroid union $\mathcal{M}^k = (E, \mathcal{I}^k)$, the simple greedy algorithm could perform very poor due to inherent non-uniformity of $\mathcal{M}^k$. In the matroid OCRS by [12], this is resolved using the idea of chain decomposition. For each level, an iterative procedure (Algorithm 2) is used to find a protection set S that includes all elements that are easily spanned by $R(\boldsymbol{x}) \cup S$. This is done by directly looking at the probability $\mathbf{Pr}[e \in \text{span}(R(\boldsymbol{x}) \cup S)]$.

Our idea is to construct a different chain decomposition based on functions $\omega_e(\cdot) : 2^E \to [0, k]$ that act as a "generalized occupancy indicator" for each element e, such that $\omega_e(\emptyset) = 0$, $\omega_e(S) = k$ if e is spanned by the set S, and we want $\omega_e(\cdot)$ to be as smooth as possible (i.e., 1-Lipschitz). For each level of the chain decomposition, we will add e to the protection set S whenever the expected occupancy $\mathbf{E}[\omega_e(R(\boldsymbol{x}) \cup S)]$ is large.

For k-uniform matroids, a simple occupancy indicator would be $\omega_e(S) = \min(k, |S|)$ (since we require its value to be between 0 and k). However, extending the definition of an occupancy function to a general k-fold matroid union introduces several challenges:

1. (**Compatibility with chain decomposition**) The most crucial part of the chain decomposition in [12] is to show the protection set S is always a proper subset of E (Lemma 11). Similarly, we will need to show that it is always possible to find a protection set $S \subsetneq E$ such that the expected occupancy $\mathbf{E}[\omega_e(R(\boldsymbol{x}) \cup S)]$ for every $e \in E \setminus S$ is smaller than k by a large enough margin.
2. (**Chernoff-strength concentration**) Based on the fact that $\mathbf{E}[\omega_e(R(\boldsymbol{x}) \cup S)]$ is sufficiently smaller than k, we ultimately want to show that $\mathbf{Pr}[\omega_e(R(\boldsymbol{x}) \cup S) = k]$ is very small, which would imply a good selectability for e. This is simple for k-uniform matroids by using Chernoff bound. However, it turns out $\omega_e(\cdot)$ for general k-fold matroid unions does not admit a standard Chernoff-strength concentration inequality, and much more efforts are required to achieve a similar selectability guarantee.

4.3 An OCRS for k-fold matroid unions

In Subsubsection 4.3.1, we define our candidate occupancy functions and show some useful properties. Then, in Subsubsection 4.3.2, we show these functions are compatible with the chain decomposition approach and can be used to get an OCRS for k-fold matroid unions.

Finally, in Subsubsection 4.3.3, we prove the selectability of this OCRS by showing these functions concentrates well enough using Theorem 3. Some proofs in the section are deferred to Appendix B for ease of reading.

4.3.1 The occupancy function

To define the occupancy function, we will instead work with the following *extended k-fold unions* which essentially introduces k parallel copies for each element. They are still matroids, and OCRS for them implies OCRS for k-fold matroid unions. Therefore, it suffices for us to give an OCRS for the extended k-fold union.

▶ **Definition 12** (Extended k-fold union)**.** *Given a matroid $\mathcal{M} = (E, \mathcal{I})$ and an integer $k \geq 1$, let $\mathcal{M}_* = (E_*, \mathcal{I}_*)$ be the matroid that contains k parallel copies $(e,1), \ldots, (e,k)$ of each element $e \in E$. Formally,*

$$E_* = E \times [k] = \{(e,i) : e \in E, i \in [k]\},$$
$$\mathcal{I}_* = \{\{(e_1,i_1), \ldots, (e_t,i_t)\} : \{e_1, \ldots, e_t\} \in \mathcal{I}, i_1, \ldots, i_t \in [k]\}.$$

And we define the extended k-fold union $\mathcal{M}_*^k = (E_*, \mathcal{I}_*^k)$ *of $\mathcal{M}$ to be the k-fold union of $\mathcal{M}_*$.*

▶ **Lemma 13.** *The extended k-fold union $\mathcal{M}_*^k$ of a matroid $\mathcal{M}$ is a matroid. Furthermore, a c-selectable OCRS for $\mathcal{M}_*^k$ implies a c-selectable OCRS for $\mathcal{M}^k$.*

We are now ready to define the following *occupancy function* on $\mathcal{M}_*^k = (E_* = E \times [k], \mathcal{I}_*^k)$. Intuitively, the function indicates the number of "slots" for elements $(e,\cdot) \in E_*$ that are occupied by the elements in S. We then show the occupancy function has good properties: it is monotone and 1-Lipschitz. More importantly, the value of $\omega_e(S)$ can be used to deduce whether $(e,\cdot) \in E_*$ is spanned by other elements in S.

▶ **Definition 14** (Occupancy function)**.** *Given an extended k-fold union $\mathcal{M}_*^k = (E_* = E \times [k], \mathcal{I}_*^k)$, for every $e \in E$, define its* occupancy function $\omega_e : 2^{E_*} \to [0,k]$ *as the function where for all $S \subseteq E_*$,*[7]

$$\omega_e(S) = k - \operatorname{rank}(S \cup (\{e\} \times [k])) + \operatorname{rank}(S).$$

▶ **Lemma 15.** *For any extended k-fold union $\mathcal{M}_*^k = (E_*, \mathcal{I}_*^k)$ and element $(e,i) \in E_*$, ω_e satisfies*

1. (Monotone) *$\omega_e(S) \leq \omega_e(T)$ for every $S \subseteq T \subseteq E_*$;*
2. (1-Lipschitz) *$\omega_e(S \cup \{a\}) - \omega_e(S) \leq 1$ for every $S \subseteq E_*$ and $a \in E_*$.*

▶ **Lemma 16.** *For any extended k-fold union $\mathcal{M}_*^k = (E_*, \mathcal{I}_*^k)$, element $(e,i) \in E_*$, and set $S \subseteq E_*$, $\omega_e(S) < k$ implies $(e,i) \notin \operatorname{span}(S \setminus \{(e,i)\})$.*

▶ **Example 17.** When $\mathcal{M}$ is a 1-uniform matroid of size n, its extended k-fold union $\mathcal{M}_*^k$ is a k-uniform matroid of size kn. For every $e \in E$ and $S \subseteq E_*$, we have

$$\operatorname{rank}(S \cup (\{e\} \times [k])) = \min(k, |S \cup (\{e\} \times [k])|) = k,$$
$$\operatorname{rank}(S) = \min(k, |S|).$$

Therefore, $\omega_e(S) = \min(k, |S|)$, i.e., the number of occupied slots by S.

Also, note that for any $\boldsymbol{x_*} \in (1 - O(\sqrt{\frac{\log k}{k}})) \cdot \mathcal{P}_{\mathcal{M}_*^k}$, the value $\omega_e(R(\boldsymbol{x_*}))$ concentrates very well as a capped sum over Bernoulli random variables. Therefore, the bad event $\omega_e(R(\boldsymbol{x_*})) = k$ rarely happens and the simple greedy algorithm without protection works.

[7] When it is clear from context, we will use rank(·)/span(·) to denote the rank/span of a set of elements in $\mathcal{M}_*^k$ for the ease of notation.

4.3.2 Chain decomposition based on occupancy functions

Similar to the matroid OCRS by [12], our OCRS for extended k-fold union $\mathcal{M}_*^k = (E_*, \mathcal{I}_*^k)$ and $\boldsymbol{x}_* \in b \cdot \mathcal{P}_{\mathcal{M}_*^k}$ starts with an offline prepossessing step that finds the following chain decomposition of elements in E_*,

$$\emptyset = N_\ell \subsetneq N_{\ell-1} \subsetneq \cdots \subsetneq N_1 \subsetneq N_0 = E_*$$

where $N_{j+1} = \textsc{KFoldProtect}(\mathcal{M}_*^k|_{N_j}, \boldsymbol{x}_*|_{N_j}, b)$ for every $0 \le j < \ell$, as described in Algorithm 3. Unlike Algorithm 2, it relies on the occupancy functions which are only defined for extended k-fold unions.

Algorithm 3 Find a protection set S for extended k-fold union $\mathcal{M}_*^k = (E_*, \mathcal{I}_*^k)$ and $\boldsymbol{x}_* \in b \cdot \mathcal{P}_{\mathcal{M}_*^k}$.

function $\textsc{KFoldProtect}(\mathcal{M}_*^k, \boldsymbol{x}_*, b)$
  $S_0, S \leftarrow \emptyset$
  while $\exists e \in E \setminus S_0,\ \mathbf{E}[\omega_e(R(\boldsymbol{x}_*) \cup S)] > bk$ **do**
    $S_0 \leftarrow S_0 \cup \{e\}$
    $S \leftarrow S \cup (\{e\} \times [k])$
  return S

Before introducing our OCRS, we need to make sure the chain decomposition above is well-defined, i.e., Algorithm 3 will always returns a proper subset S of elements, and $\mathcal{M}_*^k|_S$ remains an extended k-fold union. This is formally stated in Lemma 18, which resembles Lemma 11 in [12].

▶ **Lemma 18.** *For any $b \in (0,1)$, any extended k-fold union $\mathcal{M}_*^k = (E_*, \mathcal{I}_*^k)$ and $\boldsymbol{x}_* \in b \cdot \mathcal{P}_{\mathcal{M}_*^k}$, $S \subsetneq E_*$ for $S = \textsc{KFoldProtect}(\mathcal{M}_*^k, \boldsymbol{x}_*, b)$. Moreover, $\mathcal{M}_*^k|_S$ remains an extended k-fold union.*

Having obtained such a chain decomposition for $\mathcal{M}_*^k$ and $\boldsymbol{x}_* \in b \cdot \mathcal{P}_{\mathcal{M}_*^k}$, our OCRS is simply running the modified greedy algorithm, Algorithm 1, for each submatroid $\mathcal{M}_*^k|_{N_j}$ with a protection set N_{j+1} for all $0 \le j < \ell$ together. Note that although the chain decomposition is constructed with $\boldsymbol{x}_*$, an extra scaling factor of $e^{-(1-b)}$ will be applied before invoking Algorithm 1. This will be useful later when we apply the bicriterion concentration inequality.

Algorithm 4 OCRS for extended k-fold union $\mathcal{M}_*^k = (E_*, \mathcal{I}_*^k)$ and $\boldsymbol{x}_* \in b \cdot \mathcal{P}_{\mathcal{M}_*^k}$.

Construct the chain decomposition $\emptyset = N_\ell \subsetneq \cdots \subsetneq N_1 \subsetneq N_0 = E_*$ for $\mathcal{M}_*^k$ and $\boldsymbol{x}_*$
for each arriving active element $(e, i) \in E_*$ **do**
  Sample $r \sim \mathrm{Ber}(e^{-(1-b)})$
  if $r = 0$ **then**
    Reject (e, i)
  else
    ▷ *The set of remaining active elements follows the same distribution as $R(e^{-(1-b)}\boldsymbol{x}_*)$.* ◁
    Find $0 \le j < \ell$ such that $(e, i) \in N_j \setminus N_{i+1}$
    Invoke Algorithm 1 for $\mathcal{M}_*^k|_{N_j}$ with protection set N_{j+1} for (e, i)

The feasibility of such a scheme follows exactly from [12] as running Algorithm 1 on any chain decomposition always produces an independent set. We are left to show the OCRS guarantees a good selectability for any $\mathcal{M}_*^k$ and $\boldsymbol{x}_* \in b \cdot \mathcal{P}_{\mathcal{M}_*^k}$ for some parameter b. In fact, we will set $b = 1 - \sqrt{\frac{\log k}{k}}$ and show the selectability is at least $1 - O(\sqrt{\frac{\log k}{k}})$, proving Theorem 9.

4.3.3 Analyzing the selectability

Without loss of generality, let us focus on the selectability of elements in the first layer $E_* \setminus N_1$, since a same proof would work for all submatroid $\mathcal{M}_*^k|_{N_j}$ as they remains to be extended k-fold unions.

By Lemma 16, for every element $(e,i) \in E_* \setminus N_1$, its selectability can be lower bounded as

$$\begin{aligned}\mathbf{Pr}[(e,i) \text{ is accepted} \mid (e,i) \text{ is active}] &\geq \mathbf{Pr}[(e,i) \notin \operatorname{span}((R(e^{-(1-b)}\boldsymbol{x}_*) \setminus \{(e,i)\}) \cup N_1)] \\ &\geq \mathbf{Pr}[\omega_e(R(e^{-(1-b)}\boldsymbol{x}_*) \cup N_1) < k].\end{aligned}$$

On the other hand, by the way chain decomposition is obtained using Algorithm 3, we know even without the extra scaling of $e^{-(1-b)}$, the expected value of $\omega_e(R(\boldsymbol{x}_*) \cup N_1)$ is not too close to k:

$$\mathbf{E}[\omega_e(R(\boldsymbol{x}_*) \cup N_1)] \leq bk.$$

For the ease of notation, denote $X = R(\boldsymbol{x}_*)$ and $X' = R(e^{-(1-b)}\boldsymbol{x}_*)$. Fixing an element $(e,i) \in E_*$, define the function $f : 2^{E_*} \to [0,k]$ where for every $S \subseteq E_*$,

$$f(S) = \omega_e(S \cup N_1).$$

Then, to lower bound selectability for (e,i), it is equivalent to upper bound $\mathbf{Pr}[f(X') = k]$ given that $\mathbf{E}[f(X)] \leq bk$. Specifically, to get a selectability of $1 - O(\sqrt{\frac{\log k}{k}})$, we will set $b = 1 - \sqrt{\frac{\log k}{k}}$, and it suffices to show the following *bicriterion concentration inequality*:

$$\mathbf{E}[f(X)] \leq k - \sqrt{k \log k} \implies \mathbf{Pr}\left[f(X') \geq \mathbf{E}[f(X)] + \sqrt{k \log k}\right] \leq O\left(\frac{1}{k}\right). \tag{$*$}$$

By Lemma 15, we know f is always monotone and 1-Lipschitz. Then, using Theorem 3 (and recall that $X' = R(e^{-\sqrt{\log k/k}}\boldsymbol{x}_*)$), we have

$$\mathbf{Pr}\left[f(X') \geq \mathbf{E}[f(X)] + \sqrt{k \log k}\right] \leq \exp\left(-\sqrt{\frac{\log k}{k}} \cdot \sqrt{k \log k}\right) = \frac{1}{k}.$$

Therefore, for extended k-fold union $\mathcal{M}_*^k$ and $\boldsymbol{x}_* \in b \cdot \mathcal{P}_{\mathcal{M}_*^k}$, running Algorithm 4 yields

$$\begin{aligned}\mathbf{Pr}[(e,i) \text{ is accepted} \mid (e,i) \text{ is active}] &\geq 1 - \mathbf{Pr}\left[f(X') \geq k\right] \\ &\geq 1 - \mathbf{Pr}\left[f(X') \geq \mathbf{E}[f(X)] + \sqrt{k \log k}\right] \geq 1 - \frac{1}{k}.\end{aligned}$$

Together with Fact 10 and Lemma 13, we prove Theorem 9 by showing the existence of an OCRS for all k-fold union $\mathcal{M}^k$ and $\boldsymbol{x}_* \in \mathcal{P}_{\mathcal{M}^k}$ with a selectability of

$$\left(1 - \frac{1}{k}\right) \cdot b \cdot e^{-(1-b)} = \left(1 - \frac{1}{k}\right) \cdot \left(1 - \sqrt{\frac{\log k}{k}}\right) \cdot e^{-\sqrt{\frac{\log k}{k}}} = 1 - O\left(\sqrt{\frac{\log k}{k}}\right).$$

▶ Remark 19. It might seems bizarre and unnecessary to consider $f(X')$ instead of $f(X)$. Indeed, since f is monotone non-decreasing, the following claim that only contains $f(X)$ would imply $(*)$, and it looks more like a standard concentration inequality:

$$\mathbf{E}[f(X)] \leq k - \sqrt{k \log k} \implies \mathbf{Pr}\left[f(X) \geq \mathbf{E}[f(X)] + \sqrt{k \log k}\right] \leq O\left(\frac{1}{k}\right). \tag{$**$}$$

We know (∗∗) is true when f is a sum over Bernoulli random variables by Chernoff bound, and it is tempting to use more powerful concentration inequalities to prove (∗∗) for general 1-Lipschitz f. Unfortunately, such a bound does not exist for general monotone and 1-Lipschitz set functions (see Section 5 for details), and it turns out to be impossible even for the specific f we use here, as Example 29 shown.

5 A Bicriterion Concentration Inequality

In this section, we assume the ground set $E = [n]$ and consider a function $f : \{0,1\}^n \to \mathbb{R}$ that satisfies the following properties:[8]

1. (**Monotone**) $f(\boldsymbol{x}) \leq f(\boldsymbol{y})$ for all $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^n$ where $\boldsymbol{x} \leq \boldsymbol{y}$ (element-wise).
2. (**1-Lipschitz**) $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq \|\boldsymbol{x} - \boldsymbol{y}\|_1$ for all $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^n$.

Also, let $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ be a vector of n independent Bernoulli random variables where $X_i \sim \mathrm{Ber}(p_i)$ for each $i \in [n]$ and $\boldsymbol{p} \in [0,1]^n$. For simplicity, we denote this as $\boldsymbol{X} \sim \mathrm{Ber}(\boldsymbol{p})$.

We are interested in how well $f(\boldsymbol{X})$ concentrates on its upper tail. By McDiarmid's inequality (Theorem 28), for every $t > 0$,

$$\mathbf{Pr}\left[f(\boldsymbol{X}) \geq \mathbf{E}[f(\boldsymbol{X})] + t\right] \leq e^{-\frac{2t^2}{n}},$$

Unfortunately, the bound depends on the dimension n, whereas our application in Section 4 requires a *dimension-free* bound that is independent from n. In fact, it is known that dimension-free concentration inequality does not exist for f in general (see, e.g., [27]).

The good news is that, for our application, it suffices to consider another $\boldsymbol{X'} \sim \mathrm{Ber}(\boldsymbol{p'})$ with slightly smaller parameters $\boldsymbol{p'} < \boldsymbol{p}$ and show $f(\boldsymbol{X'})$ does not exceed $\mathbf{E}[f(\boldsymbol{X})]$ by much, with high probability. Formally, we define $\boldsymbol{X}^{(s)}$ with a scaling factor s as follows:

▶ **Definition 20** (Scaling). *Given n independent Bernoulli random variables $\boldsymbol{X} \sim \mathrm{Ber}(\boldsymbol{p})$, for any* scaling factor *$s \geq 0$, define $\boldsymbol{X}^{(s)} \sim \mathrm{Ber}(e^{-s}\boldsymbol{p})$. In other words, $X_i^{(s)} \sim \mathrm{Ber}(e^{-s}p_i)$ for all $i \in [n]$.*

And we prove Theorem 3, a *bicriterion concentration inequality*, where the bound depends on both the scaling factor s and the deviation size t.

▶ **Theorem 3.** *Let $f : \{0,1\}^E \to \mathbb{R}$ be a monotone 1-Lipschitz function. For any $s \in (0,1]$, $t > 0$:*

$$\mathbf{Pr}\left[f(\boldsymbol{X}^{(s)}) \geq \mathbf{E}[f(\boldsymbol{X})] + t\right] \leq e^{-st}.$$

For our application in Section 4, we basically set $s = \sqrt{\frac{\log k}{k}}, t = \sqrt{k \log k}$ for some $k \approx \mathbf{E}[f(\boldsymbol{X})]$ and the inequality gives us $\mathbf{Pr}[f(\boldsymbol{X}^{(s)}) \geq k + \sqrt{k \log k}] \leq \frac{1}{k}$. Note that this bound is sharp up to a constant factor in the exponent: even in the case where $f(\boldsymbol{x}) = \sum_{i=1}^n x_i$, the Chernoff bound of $f(\boldsymbol{X}^{(s)})$ only yields $\mathbf{Pr}[f(\boldsymbol{X}^{(s)}) \geq k + \sqrt{k \log k}] \leq O(\frac{1}{k^c})$ for some constant c.

[8] Note that f can be equivalently viewed as a function over subsets of a ground set of size n, as we did in Section 4.

5.1 Technical overview

Before getting into the proof, let us first outline our approach and highlight the main difficulty. Our proof utilizes the entropy method for self-bounding functions [5, 6, 22, 7]. Roughly speaking, to prove a exponential concentration inequality for some $Z = f(\boldsymbol{X})$, the plan is to establish a differential inequality for the moment-generating function $\mathbf{E}[e^{\lambda Z}]$ based on the following modified logarithmic Sobolev inequality. If this differential inequality implies strong bounds for $\mathbf{E}[e^{\lambda Z}]$, a concentration inequality can be subsequently obtained.

▶ **Lemma 21** (A modified logarithmic Sobolev inequality [20])**.** *Given n independent Bernoulli random variables $\boldsymbol{X}$ and a function $f : \{0,1\}^n \to \mathbb{R}$. Let $Z = f(\boldsymbol{X})$ and $Z_i = f_i(X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n)$ for an arbitrary function $f_i : \{0,1\}^{n-1} \to \mathbb{R}$. For any $\lambda \in \mathbb{R}$,*

$$\lambda \mathbf{E}\left[Ze^{\lambda Z}\right] - \mathbf{E}\left[e^{\lambda Z}\right] \log \mathbf{E}\left[e^{\lambda Z}\right] \leq \sum_{i=1}^{n} \mathbf{E}\left[e^{\lambda Z} \phi(-\lambda(Z - Z_i))\right]$$

where $\phi(x) = e^x - x - 1$.

Whether Lemma 21 can be effectively converted into a useful differential inequality for $\mathbf{E}[e^{\lambda Z}]$ depends on the choice of $\{Z_i\}_{i\in[n]}$. For a monotone function f, a typical choice is $Z_i = f(X_1, \ldots, X_{i-1}, 0, X_{i+1}, \ldots, X_n)$, and previous works have demonstrated that such a conversion is possible if f is 1-Lipschitz and the following condition holds almost surely for some constants $a, b \geq 0$: [9]

$$\sum_{i=1}^{n} Z - Z_i \leq aZ + b. \tag{†}$$

Now, given $Z^{(s)} = f(\boldsymbol{X}^{(s)})$ under a scaling factor $s > 0$, one might attempt to similarly derive a differential inequality of $\mathbf{E}[e^{\lambda Z^{(s)}}]$ based on Lemma 21 if the condition (†) can be satisfied. In fact, if we define $Z_i^{(s)} = f(X_1^{(s)}, \ldots, X_{i-1}^{(s)}, 0, X_{i+1}^{(s)}, \ldots, X_n^{(s)})$, the following holds:

$$\mathbf{E}\left[\sum_{i=1}^{n} Z^{(s)} - Z_i^{(s)}\right] = -\frac{\mathrm{d}}{\mathrm{d}s} \mathbf{E}\left[Z^{(s)}\right].$$

Thus, if $-\frac{\mathrm{d}}{\mathrm{d}s} \mathbf{E}[Z^{(s)}] \leq aZ^{(s)} + b$, then (†) holds in expectation for $Z^{(s)}$; otherwise, $E[Z^{(s)}]$ is decreasing rapidly with respect to s at that point.

As a result, either there exists some $s^* \in (0, s)$ such that (†) holds *in expectation* for $Z^{(s^*)}$, or $\mathbf{E}[Z^{(s)}]$ becomes significantly smaller than $\mathbf{E}[Z^{(0)}]$. Intuitively, the latter case should directly imply a bicriterion concentration result, leaving only the former case to be addressed.[10] However, it turns out that such a use of Lemma 21 crucially depends on (†) holding *almost surely*, which is not applicable to such $Z^{(s^*)}$ in the former case.[11]

Given this limitation, rather than working with moment-generating functions directly, we propose an alternative approach. Our key idea is to relate Lemma 21 with the following unconventional function, defined for every $\lambda \geq 0$:

$$F(\lambda) = \mathbf{E}\left[e^{\lambda Z^{(\lambda)}}\right].$$

---

[9] In this case, f is a so-called (a, b)-*self-bounding* function [22, 7].

[10] If we do not aim for an exponential tail bound, these observations indeed suffice to get a Chebyshev-type bicriterion concentration inequality for $Z^{(s)}$, by using Efron-Stein inequality to bound its variance in the former case.

[11] Specifically, applying the entropy method for $\mathbf{E}[e^{\lambda Z}]$ requires $\sum_{i=1}^{n} \mathbf{E}[e^{\lambda Z}(Z - Z_i)] \leq \mathbf{E}[e^{\lambda Z}(aZ + b)]$ for every λ, which might be false even if (†) holds with very high probability.

Note that this is not a moment-generating function, as λ here also serves as the scaling factor of Z, causing the random variable $Z^{(\lambda)}$ to change with it. Surprisingly, we can obtain the following upper bound for the derivative of $F(\lambda)$ that aligns well with Lemma 21.

▶ **Lemma 22.** *Given n independent Bernoulli random variables $\boldsymbol{X}$ and a monotone 1-Lipschitz function $f : \{0,1\}^n \to \mathbb{R}$. For any $\lambda \in (0,1]$,*

$$F'(\lambda) \le \mathbf{E}\left[Z^{(\lambda)} e^{\lambda Z^{(\lambda)}}\right] - \frac{1}{\lambda}\sum_{i=1}^{n} \mathbf{E}\left[e^{\lambda Z^{(\lambda)}} \phi(-\lambda(Z^{(\lambda)} - Z_i^{(\lambda)}))\right]$$

where $Z^{(\lambda)} = f(\boldsymbol{X}^{(\lambda)})$, $Z_i^{(\lambda)} = f(X_1^{(\lambda)}, \ldots, X_{i-1}^{(\lambda)}, 0, X_{i+1}^{(\lambda)}, \ldots, X_n^{(\lambda)})$, and $F(\lambda) = \mathbf{E}[e^{\lambda Z^{(\lambda)}}]$.

By combining Lemma 22 with Lemma 21, we can conclude that for all $\lambda \in (0,1]$,

$$\lambda F'(\lambda) \le F(\lambda) \log F(\lambda).$$

Solving this differential inequality provides an upper bound for $F(\lambda)$. Theorem 3 then follows by applying Markov's inequality to the random variable $e^{sZ^{(s)}}$.

5.2 Proof of Theorem 3

Let $Z^{(\lambda)} = f(\boldsymbol{X}^{(\lambda)})$ and $Z_i^{(\lambda)} = f(X_1^{(\lambda)}, \ldots, X_{i-1}^{(\lambda)}, 0, X_{i+1}^{(\lambda)}, \ldots, X_n^{(\lambda)})$ throughout the proof. Given Lemma 21 and Lemma 22, it is not hard to show the bicriterion concentration inequality.

Proof of Theorem 3. For any $\lambda > 0$, we apply Lemma 21 to $Z^{(\lambda)}$ and $\{Z_i^{(\lambda)}\}_{i\in[n]}$ and obtain

$$\lambda \mathbf{E}\left[Z^{(\lambda)} e^{\lambda Z^{(\lambda)}}\right] - \mathbf{E}\left[e^{\lambda Z^{(\lambda)}}\right] \log \mathbf{E}\left[e^{\lambda Z^{(\lambda)}}\right] \le \sum_{i=1}^{n} \mathbf{E}\left[e^{\lambda Z^{(\lambda)}} \phi(-\lambda(Z^{\lambda} - Z_i^{(\lambda)}))\right]$$

where $\phi(x) = e^x - x - 1$. Rearranging the inequality, we have

$$\lambda\left(\mathbf{E}\left[Z^{(\lambda)} e^{\lambda Z^{(\lambda)}}\right] - \frac{1}{\lambda}\sum_{i=1}^{n} \mathbf{E}\left[e^{\lambda Z^{(\lambda)}} \phi(-\lambda(Z^{\lambda} - Z_i^{(\lambda)}))\right]\right) \le \mathbf{E}\left[e^{\lambda Z^{(\lambda)}}\right] \log \mathbf{E}\left[e^{\lambda Z^{(\lambda)}}\right].$$

Together with Lemma 22, this gives us the following differential inequality for $F(\lambda) = \mathbf{E}[e^{\lambda Z^{(\lambda)}}]$:

$$\lambda F'(\lambda) \le F(\lambda) \log F(\lambda), \quad \forall \lambda \in (0,1].$$

And by letting $G(\lambda) = \log F(\lambda)$, we can rewrite the inequality as

$$\lambda G'(\lambda) \le G(\lambda), \quad \forall \lambda \in (0,1].$$

Note that $G_0(\lambda) = \lambda \mathbf{E}[Z^{(0)}]$ is a solution to $\lambda G'(\lambda) = G(\lambda)$ for $\lambda \in (0,1]$. Define $g(\lambda) = \frac{G(\lambda) - G_0(\lambda)}{\lambda}$ and we have

$$g'(\lambda) = \frac{G'(\lambda) - G_0'(\lambda)}{\lambda} - \frac{G(\lambda) - G_0(\lambda)}{\lambda^2} = \frac{(\lambda G'(\lambda) - G(\lambda)) - (\lambda G_0'(\lambda) - G_0(\lambda))}{\lambda^2} \le 0.$$

Also note that $\lim_{\lambda\to 0^+} \frac{G(\lambda)}{\lambda} = G'(0) = \frac{F'(0)}{F(0)} = \mathbf{E}[Z^{(0)}]$ (where the last equality holds by Lemma 23), therefore $\lim_{\lambda\to 0^+} g(\lambda) = 0$. Combining this with $g' \le 0$, we conclude that g is non-positive on $(0,1]$. In other words, for $\lambda \in (0,1]$,

$$G(\lambda) \le G_0(\lambda) = \lambda \mathbf{E}[Z^{(0)}].$$

Finally, by Markov's inequality, we conclude that for any $\lambda \in (0,1]$ and $t > 0$,

$$\mathbf{Pr}\left[Z^{(\lambda)} \geq \mathbf{E}[Z^{(0)}] + t\right] = \mathbf{Pr}\left[e^{\lambda Z^{(\lambda)}} \geq e^{\lambda(\mathbf{E}[Z^{(0)}]+t)}\right] \leq \frac{\mathbf{E}[e^{\lambda Z^{(\lambda)}}]}{e^{\lambda(\mathbf{E}[Z^{(0)}]+t)}} \leq e^{-\lambda t}. \quad \blacktriangleleft$$

We are left to prove Lemma 22. Let us first compute $F'(\lambda)$ by definition.

▶ **Lemma 23.** *For any $\lambda \geq 0$, $F'(\lambda) = \mathbf{E}[Z^{(\lambda)} e^{\lambda Z^{(\lambda)}}] - \sum_{i=1}^{n} \mathbf{E}[e^{\lambda Z^{(\lambda)}} - e^{\lambda Z_i^{(\lambda)}}]$.*

Proof. Define a function $h : \mathbb{R} \times [0,1]^n \to \mathbb{R}$ as

$$h(t, \boldsymbol{q}) = \mathop{\mathbf{E}}_{\boldsymbol{Y} \sim \mathrm{Ber}(\boldsymbol{q})}\left[e^{tf(\boldsymbol{Y})}\right].$$

For each $i \in [n]$ and $b \in \{0,1\}$, denote $f_{i,b}(\boldsymbol{Y}) = f(Y_1, \ldots, Y_{i-1}, b, Y_{i+1}, \ldots, Y_n)$ and we can compute the partial derivative of h with respect to q_i as

$$\begin{aligned}\frac{\partial}{\partial q_i} h(t, \boldsymbol{q}) &= \frac{\partial}{\partial q_i}\left(q_i \mathop{\mathbf{E}}_{\boldsymbol{Y} \sim \mathrm{Ber}(\boldsymbol{q})}[e^{tf_{i,1}(\boldsymbol{Y})}] + (1 - q_i) \mathop{\mathbf{E}}_{\boldsymbol{Y} \sim \mathrm{Ber}(\boldsymbol{q})}[e^{tf_{i,0}(\boldsymbol{Y})}]\right) \\ &= \mathop{\mathbf{E}}_{\boldsymbol{Y} \sim \mathrm{Ber}(\boldsymbol{q})}[e^{tf_{i,1}(\boldsymbol{Y})} - e^{tf_{i,0}(\boldsymbol{Y})}].\end{aligned}$$

Recall that $\boldsymbol{X} \sim \mathrm{Ber}(\boldsymbol{p})$ and $F(\lambda) = \mathbf{E}[e^{\lambda Z^{(\lambda)}}] = h(\lambda, e^{-\lambda}\boldsymbol{p})$. Therefore,

$$\begin{aligned}F'(\lambda) &= \frac{\mathrm{d}t}{\mathrm{d}\lambda} \cdot \frac{\partial}{\partial t} h(\lambda, e^{-\lambda}\boldsymbol{p}) + \sum_{i=1}^{n} \frac{\mathrm{d}q_i}{\mathrm{d}\lambda} \cdot \frac{\partial}{\partial q_i} h(\lambda, e^{-\lambda}\boldsymbol{p}) \\ &= 1 \cdot \mathbf{E}\left[f(\boldsymbol{X}^{(\lambda)}) e^{\lambda f(\boldsymbol{X}^{(\lambda)})}\right] + \sum_{i=1}^{n} (-e^{-\lambda} p_i) \cdot \mathop{\mathbf{E}}_{\boldsymbol{Y} \sim \mathrm{Ber}(e^{-\lambda}\boldsymbol{p})}\left[e^{\lambda f_{i,1}(\boldsymbol{Y})} - e^{\lambda f_{i,0}(\boldsymbol{Y})}\right] \\ &= \mathbf{E}\left[f(\boldsymbol{X}^{(\lambda)}) e^{\lambda f(\boldsymbol{X}^{(\lambda)})}\right] - \sum_{i=1}^{n} \mathop{\mathbf{E}}_{\boldsymbol{Y} \sim \mathrm{Ber}(e^{-\lambda}\boldsymbol{p})}\left[e^{\lambda f(\boldsymbol{Y})} - e^{\lambda f_{i,0}(\boldsymbol{Y})}\right] \\ &= \mathbf{E}\left[Z^{(\lambda)} e^{\lambda Z^{(\lambda)}}\right] - \sum_{i=1}^{n} \mathbf{E}\left[e^{\lambda Z^{(\lambda)}} - e^{\lambda Z_i^{(\lambda)}}\right]. \quad \blacktriangleleft\end{aligned}$$

Then we further derive a lower bound to the latter term, $\sum_{i=1}^{n} \mathbf{E}[e^{\lambda Z^{(\lambda)}} - e^{\lambda Z_i^{(\lambda)}}]$.

▶ **Lemma 24.** *For any $\lambda \geq 0$, $\sum_{i=1}^{n} \mathbf{E}[e^{\lambda Z^{(\lambda)}} - e^{\lambda Z_i^{(\lambda)}}] \geq \lambda e^{-\lambda} \sum_{i=1}^{n} \mathbf{E}[e^{\lambda Z^{(\lambda)}}(Z^{(\lambda)} - Z_i^{(\lambda)})]$.*

Proof. We prove the inequality for each term separately and without expectation. For any $i \in [n]$, note that

$$e^{\lambda Z^{(\lambda)}} - e^{\lambda Z_i^{(\lambda)}} = e^{\lambda Z_i^{(\lambda)}}(e^{\lambda(Z^{(\lambda)} - Z_i^{(\lambda)})} - 1) \geq e^{\lambda Z_i^{(\lambda)}} \cdot \lambda(Z^{(\lambda)} - Z_i^{(\lambda)})$$

since $e^x - 1 \geq x$. Meanwhile, we know $Z_i^{(\lambda)} \geq Z^{(\lambda)} - 1$ by 1-Lipschitzness of f. Therefore,

$$e^{\lambda Z^{(\lambda)}} - e^{\lambda Z_i^{(\lambda)}} \geq \lambda e^{-\lambda} \cdot e^{\lambda Z^{(\lambda)}}(Z^{(\lambda)} - Z_i^{(\lambda)}). \quad \blacktriangleleft$$

The following two facts of the function $\phi(x) = e^x - x - 1$ will also be used.

▶ **Fact 25.** *For any $\lambda \in (0,1]$, $\frac{\phi(-\lambda)}{\lambda} \leq \lambda e^{-\lambda}$.*

▶ **Fact 26.** *For any $\lambda \in \mathbb{R}$ and $x \in [0,1]$, $\phi(-\lambda x) \leq \phi(-\lambda)x$.*

Now we are ready to prove the lemma.

Proof of Lemma 22. We upper bound $F'(\lambda)$ step-by-step as follows:

$$F'(\lambda) = \mathbf{E}[Z^{(\lambda)}e^{\lambda Z^{(\lambda)}}] - \sum_{i=1}^{n} \mathbf{E}\left[e^{\lambda Z^{(\lambda)}} - e^{\lambda Z_i^{(\lambda)}}\right] \qquad \text{(Lemma 23)}$$

$$\leq \mathbf{E}[Z^{(\lambda)}e^{\lambda Z^{(\lambda)}}] - \lambda e^{-\lambda} \sum_{i=1}^{n} \mathbf{E}\left[e^{\lambda Z^{(\lambda)}}(Z^{(\lambda)} - Z_i^{(\lambda)})\right] \qquad \text{(Lemma 24)}$$

$$\leq \mathbf{E}[Z^{(\lambda)}e^{\lambda Z^{(\lambda)}}] - \frac{\phi(-\lambda)}{\lambda} \sum_{i=1}^{n} \mathbf{E}\left[e^{\lambda Z^{(\lambda)}}(Z^{(\lambda)} - Z_i^{(\lambda)})\right] \qquad \text{(Fact 25)}$$

$$\leq \mathbf{E}[Z^{(\lambda)}e^{\lambda Z^{(\lambda)}}] - \frac{1}{\lambda} \sum_{i=1}^{n} \mathbf{E}\left[e^{\lambda Z^{(\lambda)}}\phi(-\lambda(Z^{(\lambda)} - Z_i^{(\lambda)}))\right] \qquad \text{(Fact 26)}$$

where in the last step we also use the fact that $Z^{(\lambda)} - Z_i^{(\lambda)} \in [0, 1]$, as f is monotone and 1-Lipschitz. ◀

References

1. Saeed Alaei. Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM Journal on Computing*, 43(2):930–972, 2014. doi:10.1137/120878422.
2. Saeed Alaei, Hu Fu, Nima Haghpanah, Jason Hartline, and Azarakhsh Malekian. Bayesian optimal auctions via multi- to single-agent reduction. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, page 17, New York, NY, USA, 2012. Association for Computing Machinery. doi:10.1145/2229012.2229017.
3. C. J. Argue, Anupam Gupta, Marco Molinaro, and Sahil Singla. Robust secretary and prophet algorithms for packing integer programs. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 1273–1297. SIAM, SIAM, 2022. doi:10.1137/1.9781611977073.53.
4. Pablo Daniel Azar, Robert Kleinberg, and S. Matthew Weinberg. Prophet inequalities with limited information. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1358–1377, 2014. doi:10.1137/1.9781611973402.100.
5. Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. A sharp concentration inequality with applications. *Random Structures & Algorithms*, 16(3):277–292, 2000. doi:10.1002/(SICI)1098-2418(200005)16:3\%3C277::AID-RSA4\%3E3.0.CO;2-1.
6. Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. Concentration inequalities using the entropy method. *The Annals of Probability*, 31(3):1583–1614, 2003.
7. Stephane Boucheron, Gabor Lugosi, and Pascal Massart. On concentration of self-bounding functions. *Electronic Journal of Probability*, 14:1884–1899, 2009.
8. Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 311–320. ACM, 2010. doi:10.1145/1806689.1806733.
9. Chandra Chekuri, Junkai Song, and Weizhong Zhang. Contention resolution for the l-fold union of a matroid via the correlation gap. In *2024 Symposium on Simplicity in Algorithms (SOSA)*, pages 396–405. SIAM, 2024. doi:10.1137/1.9781611977936.36.
10. Atanas Dinev and S. Matthew Weinberg. Simple and Optimal Online Contention Resolution Schemes for k-Uniform Matroids. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, volume 287 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 39:1–39:23, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2024.39.

11 Paul Dütting, Michal Feldman, Thomas Kesselheim, and Brendan Lucier. Prophet inequalities made easy: Stochastic optimization by pricing nonstochastic inputs. *SIAM J. Comput.*, 49(3):540–582, 2020. doi:10.1137/20M1323850.

12 Moran Feldman, Ola Svensson, and Rico Zenklusen. Online contention resolution schemes. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1014–1033. SIAM, 2016. doi:10.1137/1.9781611974331.ch72.

13 Mohammad Taghi Hajiaghayi, Robert Kleinberg, and Tuomas Sandholm. Automated online mechanism design and prophet inequalities. In *AAAI*, volume 7, pages 58–65, 2007. URL: http://www.aaai.org/Library/AAAI/2007/aaai07-009.php.

14 Jiashuo Jiang, Will Ma, and Jiawei Zhang. Tight guarantees for multi-unit prophet inequalities and online stochastic knapsack. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1221–1246. SIAM, 2022. doi:10.1137/1.9781611977073.51.

15 Thomas Kesselheim, Andreas Tönnis, Klaus Radke, and Berthold Vöcking. Primal beats dual on online packing lps in the random-order model. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 303–312, 2014. doi:10.1145/2591796.2591810.

16 Robert Kleinberg and S. Matthew Weinberg. Matroid prophet inequalities. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 123–136, 2012. doi:10.1145/2213977.2213991.

17 Ulrich Krengel and Louis Sucheston. On semiamarts, amarts, and processes with finite value. *Advances in Probability and Related Topics*, 4:197–266, 1978.

18 Felix Lazebnik, Vasiliy A Ustimenko, and Andrew J Woldar. A new series of dense graphs of high girth. *Bulletin of the American mathematical society*, 32(1):73–79, 1995.

19 Euiwoong Lee and Sahil Singla. Optimal Online Contention Resolution Schemes via Ex-Ante Prophet Inequalities. In *26th Annual European Symposium on Algorithms (ESA 2018)*, volume 112, pages 57:1–57:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ESA.2018.57.

20 Pascal Massart. About the constants in talagrand's concentration inequalities for empirical processes. *The Annals of Probability*, 28(2):863–884, 2000.

21 Colin McDiarmid et al. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188, 1989.

22 Colin McDiarmid and Bruce Reed. Concentration for self-bounding functions and an inequality of talagrand. *Random Structures & Algorithms*, 29(4):549–557, 2006. doi:10.1002/rsa.20145.

23 Aviad Rubinstein and Sahil Singla. Combinatorial prophet inequalities. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1671–1687, 2017. doi:10.1137/1.9781611974782.110.

24 Aviad Rubinstein and S. Matthew Weinberg. Simple mechanisms for a subadditive buyer and applications to revenue monotonicity. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*, pages 377–394, 2015. doi:10.1145/2764468.2764510.

25 Ester Samuel-Cahn. Comparison of threshold stop rules and maximum for independent nonnegative random variables. *Annals of Probability*, 12(4):1213–1216, 1984.

26 Gideon Schechtman. Concentration, results and applications. In *Handbook of the geometry of Banach spaces*, volume 2, pages 1603–1634. Elsevier, 2003.

27 Jan Vondrák. A note on concentration of submodular functions. *CoRR*, abs/1005.2791, 2010. doi:10.48550/arXiv.1005.2791.

A Useful Concentration Inequalities

▶ **Theorem 27** (Multiplicative Chernoff bound). *Given n independent Bernoulli random variables $X_1, X_2, \ldots, X_n$, let $X = \sum_{i=1}^{n} X_i$ denote their sum. For any $\delta > 0$, we have*

$$\mathbf{Pr}[X \geq (1+\delta)\,\mathbf{E}[X]] \leq \exp\left(-\frac{\delta^2\,\mathbf{E}[X]}{2+\delta}\right).$$

▶ **Theorem 28** (McDiarmid's inequality). *Given n independent random variables $X_1, X_2, \ldots, X_n \in \mathcal{X}$ and a function $f : \mathcal{X}^n \to \mathbb{R}$. If for every $i \in [n]$ and $x_1, x_2, \ldots, x_n, x_i' \in \mathcal{X}$, the function f satisfies*

$$|f(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n) - f(x_1, \ldots, x_{i-1}, x_i', x_{i+1}, \ldots, x_n)| \leq c_i,$$

then for any $t > 0$, we have

$$\mathbf{Pr}[f(X) \geq \mathbf{E}[f(X)] + t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^{n} c_i^2}\right).$$

B Missing Proofs and Examples

Proof of Lemma 13. It is straightforward to check $\mathcal{M}_*$ in Definition 12 is a matroid, and hence its k-fold union $\mathcal{M}_*^k$ remains a matroid by the closure property of matroid union. Also, note that the restriction of $\mathcal{M}_*^k$ to $E \times \{1\}$, $\mathcal{M}_*^k\,|_{E\times\{1\}}$, is isomorphic to the k-fold union $\mathcal{M}^k$ of $\mathcal{M}$, as there exists a simple bijection $(e, 1) \mapsto e$ between $E \times \{1\}, \mathcal{I}_*|_{E\times\{1\}}$ and $E, \mathcal{I}$. Therefore, an α-selectable OCRS for $\mathcal{M}_*^k$ can also be used as an α-selectable OCRS for $\mathcal{M}^k$. ◀

Proof of Lemma 15. Note that the rank function for any matroid is a submodular function. Therefore, $\mathrm{rank}(S \cup (\{e\} \times [k])) - \mathrm{rank}(S) \geq \mathrm{rank}(T \cup (\{e\} \times [k])) - \mathrm{rank}(T)$ for every $S \subseteq T$ by a simple induction, and thus $\omega_e(\cdot)$ is monotone.

Also, we know the rank function is monotone, and the rank of a set can increase by at most 1 after adding an element. Therefore, $\mathrm{rank}(S \cup \{a\}) \geq \mathrm{rank}(S)$ and $\mathrm{rank}(S \cup \{a\} \cup (\{e\} \times [k])) \leq \mathrm{rank}(S \cup (\{e\} \times [k])) + 1$ for every $a \in E_*$ and $S \subseteq E_*$. As a result, $\omega_e(\cdot)$ is 1-Lipschitz. ◀

Proof of Lemma 16. When $\omega_e(S) < k$, we have $\mathrm{rank}(S \cup (\{e\} \times [k])) > \mathrm{rank}(S)$ and there exists (at least) one element $(e, j) \in \{e\} \times [k]$ such that $(e, j) \notin \mathrm{span}(S)$. By definition of extended k-fold union, it further implies $(e, i) \notin \mathrm{span}(S \setminus \{(e, i)\})$. ◀

Proof of Lemma 18. It is straightforward to see $S = S_0 \times [k]$ when the algorithm terminates, and thus $\mathcal{M}_*^k|_{S_0\times[k]}$ is the extended k-fold union of $\mathcal{M}|_{S_0}$ by definition. It remains to prove $S \subsetneq E_*$. Since S must be a subset of the universe E_*, it suffices to show $S \neq E_*$. Our plan is to show that S is not full rank in $\mathcal{M}_*^k$, even after combined with all active elements $R(\boldsymbol{x}_*)$ and take the expectation, i.e.,

$$\mathbf{E}[\mathrm{rank}_{\mathcal{M}_*^k}(R(\boldsymbol{x}_*) \cup S)] < \mathrm{rank}_{\mathcal{M}_*^k}(E_*).$$

This would directly imply $S \neq E_*$ by the monotonicity of the rank function.

Denote $r = \mathrm{rank}_{\mathcal{M}}(S_0)$. Let $e_1, e_2, \ldots, e_r \in S_0$ be the elements from $\mathcal{M}$ that increase the rank of S_0 in $\mathcal{M}$ during the execution of Algorithm 3, and denote e_i (for $1 \leq i \leq r$) as the specific element that increases $\mathrm{rank}_{\mathcal{M}}(S_0)$ from $i-1$ to i. By definition, $\mathrm{span}_{\mathcal{M}}(\{e_1, e_2, \ldots, e_r\}) = S_0$. In fact, we also have

$$\mathrm{span}_{\mathcal{M}_*^k}(\{e_1, e_2 \ldots, e_r\} \times [k]) = S.$$

This is because $\{e_1, e_2, \ldots, e_r\} \times [k] \subseteq S$ is an independent set of size kr in $\mathcal{M}_*^k$ by definition of the extended k-fold union, and we can further show it is a basis of S. Suppose it is not, then there must be another independent set $T \subseteq S$ of size larger than kr. Since one can partition T into k disjoint independent sets $T_1, T_2, \ldots, T_k$ in $\mathcal{M}_*$ where $\sum_{j\in[k]} |T_j| > kr$, we know there exists some T_j of size larger than r, which leads to a contradiction as $T_j \subseteq S_0 \times [k]$ and $\operatorname{rank}_{\mathcal{M}_*}(S_0 \times [k]) = r$.

Also, by the way the algorithm picks elements to be added to S_0, for every $1 \le i \le r$ we have

$$\mathbf{E}[\omega_{e_i}(R(\boldsymbol{x}_*) \cup (\{e_1, e_2, \ldots, e_{i-1}\} \times [k]))] > bk.$$

Equivalently, we have

$$\mathbf{E}[\operatorname{rank}_{\mathcal{M}_*^k}(R(\boldsymbol{x}_*) \cup (\{e_1, e_2, \ldots, e_i\} \times [k])) - \operatorname{rank}_{\mathcal{M}_*^k}(R(\boldsymbol{x}_*) \cup (\{e_1, e_2, \ldots, e_{i-1}\} \times [k]))] < (1-b)k.$$

Together with these observations, we can upper bound $\mathbf{E}[\operatorname{rank}_{\mathcal{M}_*^k}(R(\boldsymbol{x}_*) \cup S)]$ by a telescoping sum as follows:

$$\begin{aligned}
\mathbf{E}[\operatorname{rank}_{\mathcal{M}_*^k}(R(\boldsymbol{x}_*) \cup S)] &= \mathbf{E}[\operatorname{rank}_{\mathcal{M}_*^k}(R(\boldsymbol{x}_*) \cup \{e_1, e_2, \ldots, e_r\} \times [k])] \\
&= \mathbf{E}[\operatorname{rank}_{\mathcal{M}_*^k}(R(\boldsymbol{x}_*))] + \sum_{i=1}^{r} \mathbf{E}[\operatorname{rank}_{\mathcal{M}_*^k}(R(\boldsymbol{x}_*) \cup (\{e_1, e_2, \ldots, e_i\} \times [k])) \\
&\qquad - \operatorname{rank}_{\mathcal{M}_*^k}(R(\boldsymbol{x}_*) \cup (\{e_1, e_2, \ldots, e_{i-1}\}) \times [k])] \\
&< \mathbf{E}[\operatorname{rank}_{\mathcal{M}_*^k}(R(\boldsymbol{x}_*))] + (1-b)kr.
\end{aligned}$$

The former term $\mathbf{E}[\operatorname{rank}_{\mathcal{M}_*^k}(R(\boldsymbol{x}_*))]$ can be trivially upper bounded by $\mathbf{E}[|R(\boldsymbol{x}_*)|]$ and further by $b \operatorname{rank}_{\mathcal{M}_*^k}(E_*)$ due to $\boldsymbol{x}_* \in b \cdot \mathcal{P}_{\mathcal{M}_*^k}$. For the latter term involving kr, we already know $kr = \operatorname{rank}_{\mathcal{M}_*^k}(S) \le \operatorname{rank}_{\mathcal{M}_*^k}(E_*)$. In conclusion, we have

$$\mathbf{E}[\operatorname{rank}_{\mathcal{M}_*^k}(R(\boldsymbol{x}_*) \cup S)] < b \operatorname{rank}_{\mathcal{M}_*^k}(E_*) + (1-b) \operatorname{rank}_{\mathcal{M}_*^k}(E_*) = \operatorname{rank}_{\mathcal{M}_*^k}(E_*). \qquad \blacktriangleleft$$

▶ **Example 29** (A counterexample to $(**)$)**.** Fix parameters n, k where $n \gg k$, and consider the case when $\mathcal{M}$ is an n-uniform matroid of size $2n$. Its extended k-fold union $\mathcal{M}_*^k$ is a kn-uniform matroid of size $2kn$. Similar to Example 17, for every $e \in E$ and $S \subseteq E_*$ we can derive

$$\omega_e(S) = \begin{cases} 0, & |S| \le kn - k \\ |S| - (kn-k), & kn - k < |S| < kn \\ k, & |S| \ge kn. \end{cases}$$

Since no protection is needed for uniform matroids, let $f(\cdot) = \omega_e(\cdot)$ for some fixed $e \in E$. When $\boldsymbol{x}_* = (\frac{1}{2} - \frac{1}{2n}) \cdot \mathbf{1}_{E_*}$ (namely, every element in E_* is active with probability $\frac{1}{2} - \frac{1}{2n}$), $|X|$ will follow a binomial distribution with $kn - k$ as both its mean and median. As a result,

$$\mathbf{E}[f(X)] \le k \, \mathbf{Pr}[f(X) > 0] = k \, \mathbf{Pr}[|X| > kn - k] \le \frac{k}{2},$$

$$\text{while} \quad \mathbf{Pr}[f(X) \ge k] = \mathbf{Pr}[|X| \ge kn] \ge \Omega(1), \qquad (n \gg k)$$

which is a counterexample to the claim $(**)$.

Note that this is not an actual counterexample to Algorithm 4 (even without the extra scaling) since $\boldsymbol{x}_* \notin (1 - O(\sqrt{\frac{\log k}{k}})) \cdot \mathcal{P}_{\mathcal{M}_*^k}$. But it shows that the condition $\mathbf{E}[f(X)] \le k - O(\sqrt{k \log k})$ alone is not enough to derive a good enough upper bound for $\mathbf{Pr}[f(X) \ge k]$, and it is crucial to also rely on the scaling applied to $\boldsymbol{x}_*$.

Edge-Minimum Walk of Modular Length in Polynomial Time

Antoine Amarilli
Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
LTCI, Télécom Paris, Institut polytechnique de Paris, France

Benoît Groz
Paris-Saclay University, CNRS, LISN, France

Nicole Wein
University of Michigan, Ann Arbor, MI, USA

Abstract

We study the problem of finding, in a directed graph, an *st*-walk of length r mod q which is edge-minimum, i.e., uses the smallest number of *distinct* edges. Despite the vast literature on paths and cycles with modularity constraints, to the best of our knowledge we are the first to study this problem. Our main result is a polynomial-time algorithm that solves this task when r and q are constants.

We also show how our proof technique gives an algorithm to solve a generalization of the well-known Directed Steiner Network problem, in which connections between endpoint pairs are required to satisfy modularity constraints on their length. Our algorithm is polynomial when the number of endpoint pairs and the modularity constraints on the pairs are constants.

In this version of the article, proofs and examples are omitted because of space constraints. Detailed proofs are available in the full version [3].

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Shortest paths

Keywords and phrases Directed Steiner Network, Modularity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.5

Related Version *Full Version*: `https://arxiv.org/abs/2412.01614` [3]

Funding *Antoine Amarilli*: Amarilli was partially supported by the ANR project EQUUS ANR-19-CE48-0019, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 431183758, and by the ANR project ANR-18-CE23-0003-02 ("CQFD").

Acknowledgements We are grateful to the reviewers of the conference version for their helpful feedback. We also would like to thank the Simons Institute Fall 2023 programs "Logic and Algorithms in Database Theory and AI" and "Data Structures and Optimization for Fast Algorithms" for the initiation of this work. Last, we are grateful to Xiao Hu and Mikaël Monet for early discussions.

1 Introduction

We begin with a simple question: Given an n-vertex, m-edge directed graph G and terminals s,t, can we efficiently find an odd-length *st*-walk that is "edge-minimum", i.e., has the minimum number of *distinct* edges? This question may appear similar to classical problems from the vast literature on paths and cycles with parity constraints. For instance, one may think of the classical "shortest odd *st*-path" problem, which is well-known to be NP-hard (this is via a simple reduction from the 2-disjoint paths problem of [19], and can be proved in the same way as [37, Proposition 2.1]). However, this hardness result only applies to *simple paths*, whereas the edge-minimum odd *st*-walk may not be a simple path. One may also think of the "shortest odd *st*-walk" problem (i.e., minimizing the length), which is well-known to

© Antoine Amarilli, Benoît Groz, and Nicole Wein;
licensed under Creative Commons License CC-BY 4.0

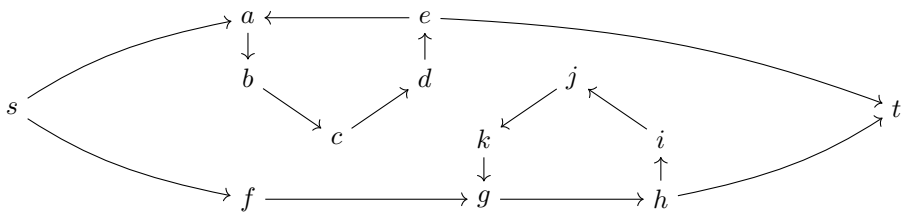


Figure 1 An example of a graph with different answers to the problems of finding the edge-minimum odd st-walk, the shortest odd st-walk, and the shortest odd simple st-path. First, every simple path from s to t is of even length, so there is no shortest odd st-path. Second, the *shortest* odd st-walk uses the bottom cycle $s, f, g, h, i, j, k, g, h, t$: it has length 9 and uses 8 distinct edges. Third, the *edge-minimum* odd st-walk uses the top cycle: $s, a, b, c, d, e, a, b, c, d, e, t$. It has length 11 but uses only 7 distinct edges.

be in polynomial time[1]. However, the *edge-minimum* odd st-walk may be *longer* than the *shortest* odd st-walk while using *fewer distinct edges*. See Figure 1 for an example of a graph in which the solutions to all three of these problems is different.

The focus of this paper is the *Edge-Minimum Walk of Modular Length* problem, which is the above problem generalized to an arbitrary modularity q and remainder r. Let us define it formally:

Edge-Minimum Walk of Modular Length (EWM):
Input: An unweighted directed[2] graph G, terminals s,t, and non-negative integers $r < q$.
Output: An st-walk of length r mod q which is edge-minimum, i.e., uses the minimum number of distinct edges (or $\emptyset$ if no such walk exists).

We stress that the modularity constraint does not apply to the number of distinct edges, but only to the length of the walk. We are most interested in the regime where q, and hence r, are constants; our algorithms will work without this assumption, but they achieve worse complexities.

Despite the vast literature on paths and cycles of given modularities [31, 26, 37, 36, 39, 27, 4, 38, 10, 41, 34, 32, 29, 40, 24, 1, 35, 6, 25, 14, 16, 7, 23] (see [2] for a survey), to the best of our knowledge we are the first to study the EWM problem.

EWM can also be viewed as a problem in the field of *network design*. Network design problems ask questions of the form "find an edge-minimum subgraph with a certain property". Famous network design problems include, for instance, Minimum Spanning Tree, Traveling Salesperson, and st-Shortest Path. We can give an equivalent rephrasing of EWM in this way: find an edge-minimum subgraph that contains an st-walk of length r mod q. The network design problem most related to EWM is the Directed Steiner Network problem (DSN) [17, 18, 8, 20, 15, 9, 33, 21, 13]. In DSN, the input is a directed graph G and a set of k terminal pairs $(s_1,t_1),\ldots,(s_k,t_k)$; the goal is to find an edge-minimum subgraph that

[1] Make two copies of the vertex set, and for every edge $u \to v$ in the original graph, add the edges $u_1 \to v_2$ and $u_2 \to v_1$. Then find the shortest walk from s_1 to t_2. This also trivially generalizes to modularities which are polynomial in n.

[2] One could also ask this question on an undirected graph. However, in this case, one unnatural aspect of the problem is that one can always traverse the same edge back and forth over and over to add any multiple of 2 to the path length. We use this observation to show in the full version [3] that EWM on undirected graphs reduces to EWM on directed graphs.

contains a path $s_i \to t_i$ for all terminal pairs. When k is arbitrary, DSN generalizes Directed Steiner Tree and is therefore NP-hard. When k is constant, Feldman and Ruhl [17] showed that DSN can be solved in polynomial time. In fact, we show in Section 8 that, in the special case where the set of terminal pairs is strongly connected, the DSN problem with constant k can be directly expressed as a special case of EWM with constant q. As a consequence, our main result will imply, as a byproduct, the known result that this special case of DSN is in polynomial time for constant k [17] (albeit with a larger exponent). We also study a problem generalizing both EWM and DSN and give an algorithm that subsumes the tractability of both problems, as we will explain later in the introduction.

EWM is also related to problems from database theory, in particular the evaluation of *regular path queries* (RPQs) on graph databases. More precisely, a graph database is a graph whose edges are labeled with symbols from a fixed alphabet, and an RPQ is a regular expression e over the alphabet. The results of the RPQ are vertex pairs (s, t) such that there is a walk from s to t (or sometimes a simple path or a trail, depending on the variant) whose edge labels from a word that matches e (see e.g. [12, 11, 5, 28]). In particular, RPQs of the form $a^r(a^q)^*$ express walks of length r mod q for constant r and q. The EWM problem then corresponds to the *smallest witness problem* [30, 22] for such queries, which is the problem of finding a sub-database of minimum size that satisfies the query. However, to the best of our knowledge there is no prior work on the smallest witness problem for RPQs enforcing modularity conditions, or indeed for RPQs in general: our work can be seen as a first step towards addressing this problem.

Our results. What is the complexity of EWM for constant q? It is unclear a priori; EWM is related to both polynomial-time solvable problems such as Shortest Modular-Length st-Walk and Directed Steiner Network, as well as NP-hard problems such as Shortest Modular-Length st-Path. Our main result is that EWM is in polynomial time for constant q:

▶ **Theorem 1.1.** *There is an algorithm solving EWM in time:* $n^{O(\log q)} \cdot 2^{O(q \log^2 q)}$.

Specifically, the exponent of n in Theorem 1.1 has reasonable constants: $7 + 3\log_2 q$, though we did not focus on optimizing them.

Generalizations. A natural generalization of EWM is to consider weighted graphs. EWM admits two natural notions of weights: (1) each edge has a *length*, and the length of a walk is the sum of the lengths of the edges, and (2) each edge (or vertex) has a *cost* and the edge-minimum walk is measured in terms of the sum of the costs of the edges (or vertices) that it traverses. We show in Section 7 that our algorithm extends to accommodate costs, and that it can also accommodate lengths provided that their values are polynomial. In the case where lengths and the value q are super-polynomial, we show that the problem is NP-complete by reduction from Subset Sum.

Inspired by Directed Steiner Network, we also extend our results in Section 8 from one walk to multiple walks. The input is a directed graph G, and k endpoint pairs along with modularity constraints: $(s_1, t_1, q_1, r_1), \ldots, (s_k, t_k, q_k, r_k)$, and the goal is to find an edge-minimum subgraph that contains a walk from s_i to t_i of length r_i mod q_i, for all i. Generalizing our main result, we show (Theorem 8.4) that this problem, which subsumes the DSN and EWM problems, also admits a polynomial-time algorithm for constant q_i values and a constant number k of endpoint pairs. Our approach to show the result focuses on the case of strongly connected solutions (in line with the connection to DSN mentioned earlier), before extending the proof to arbitrary solutions by re-using lemmas from [18].

2 Technical Overview

Our work is related to the Directed Steiner Network (DSN) problem studied by Feldman and Ruhl [17], where we must find an edge-minimum subgraph of the input directed graph which satisfies connectivity requirements. Their work shows that the DSN problem is tractable when the number of endpoint pairs in the connectivity requirements is constant. Following their work, Feldmann and Marx [18] have investigated the parameterized complexity of the DSN problem. One key idea of their approach is to show bounds on the *cutwidth* of solutions to the DSN problem. Informally, cutwidth is a parameter that limits how many edges are "cut" when arranging vertices in a well-chosen order $<$. Bounding the cutwidth of a graph ensures that we can go over vertices in the order of $<$ and that, at any cut along $<$, all edges except a constant number will be entirely to one side of the cut. (See Section 3 for the formal definition of cutwidth.)

Feldmann and Marx show that when the number of endpoint pairs is bounded by a constant k, then DSN instances always have a solution of constant *undirected* cutwidth (depending only on k). They further show that bounded-cutwidth solutions can be computed in polynomial time by dynamic programming. We follow the same framework and show that solutions to EWM also have bounded cutwidth. Then we can find the solution by an approach similar to the dynamic programming algorithm of [18], or to the token game of [17].

Thus, our main goal is to solve the purely structural problem of bounding the cutwidth of edge-minimum st-walks of length $r \bmod q$ in arbitrary directed graphs. We only focus on this goal in the rest of the technical overview. To provide intuition, we will begin by considering the case of $q = 2$, $r = 1$ (i.e. odd walks), and then outline the obstacles that arise when generalizing to arbitrary q, r.

Odd walks. Let w denote an edge-minimum odd-length st-walk, and let G_w be the subgraph of G consisting of the union of edges in the walk w. We first observe that, without loss of generality, w does not contain any even cycles: If w contained an even cycle, we could simply delete it, and w would still be of odd length, and still be edge-minimum. (To clarify, we are not referring to even cycles in G_w, but rather even cycles in the walk w itself. After deleting a cycle C from w, edges from C can still appear in G_w if they are traversed at another point of the walk w.)

Now suppose w contains two odd cycles in succession. Again, we could simply delete both cycles, and w would still be of odd length, and still be edge-minimum. So, we can suppose without loss of generality that w has either no cycles (and trivially has small cutwidth), or consists of a simple path P_1, followed by an odd cycle C, followed by a simple path P_2. We assume this latter case in the rest of the discussion, and we define C by looking at first time that the walk re-visits a previously visited vertex: this ensures that, by construction, P_1 is vertex-disjoint from C. However, the graph does not necessarily have small cutwidth, because P_2 could intersect P_1 and C in intricate ways forming an unbounded number of nested cycles. It is a priori conceivable that, e.g., a grid-like structure with high cutwidth could emerge. However, we can show that this does not happen.

First, we can observe that once P_2 leaves C, without loss of generality it does not return. This is because if P_2 leaves C at vertex a and then returns at vertex b, we can delete the subpath of P_2 from a to b, and instead traverse only the edges of C to get from a to b. This way, we can still ensure that w is of odd length since every traversal of C changes the parity of w, and we can traverse C for "free" without adding any extra edges.

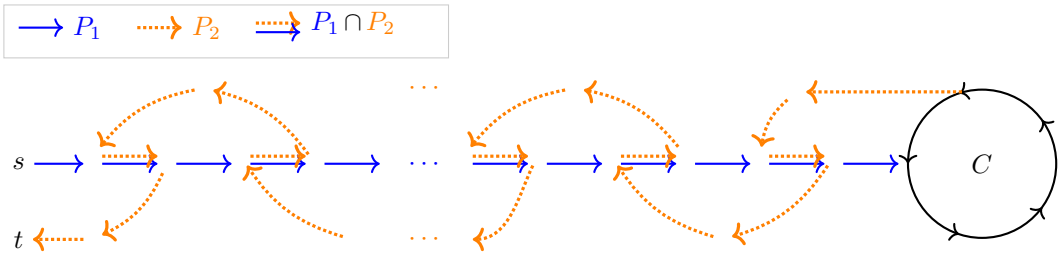


Figure 2 Specific shape of solutions to EWM for $q = 2$. Each "edge" in the figure represents a subpath, not necessarily a single edge. The cycle C is of odd length.

Finally, we consider how P_1 and P_2 can interact. We can observe that without loss of generality P_2 cannot hit a vertex a on P_1, then leave P_1, then return to a *later* vertex b on P_1. In this case we could delete the subpath of P_2 from a to b, and instead traverse P_1 to get from a to b. Again, we can still ensure that w is of odd length since we can always traverse C for free to change the parity of w. We stress that this argument only holds when P_2 re-enters P_1 at a *later* vertex, and in fact, P_2 can re-enter P_1 at an *earlier* vertex an unbounded number of times.

With all of these observations in mind, we conclude that if w has a cycle, then w must have the very specific structure depicted in Figure 2. Then, it is visually clear that any vertical cut only has a constant number of crossing edges, and thus w has constant cutwidth.

This concludes the outline for the case of odd walks. The same argument holds for even walks, and more generally an argument for a particular choice of constants q, r is easily extendable by reduction to other constant values r' with the same q, simply by adding a new source connected to the original source by a path of constant length $r' - r \bmod q$. The challenging part is extending to arbitrary q. Our approach is to define a *segment decomposition* of the walk and prove that each successive segment allows us to expand the reachable set of remainder values, so that the number of segments is bounded as a function of q (Lemma 4.3). We then show that the cutwidth of a walk can be bounded linearly in its number of segments (Proposition 5.1): we do this by defining a suitable ordering of the vertices following the *chunks* of the walk, which refine the segments in the decomposition. These two results imply that solutions to EWM have bounded cutwidth and hence that they can be found in polynomial time.

3 Preliminaries

In this section, we present preliminary notions used throughout the proof of our main result (Theorem 1.1).

Basic definitions. All graphs in the paper are finite and directed. Graphs are not necessarily simple; they may contain self-loops, but they do not contain multi-edges. Given a directed graph $G = (V, E)$, a *walk* w in G is a sequence of edges $w[1], \ldots, w[\ell]$ such that the target vertex of $w[i]$ is equal to the source vertex of $w[i+1]$ for each $1 \leq i < \ell$. Note that an edge e of G may occur at several positions in G: considering a position i for which $w[i] = e$, we write $w[i]$ to refer to that occurrence of e at the i-th position of w. However, when convenient we abuse notation and identify $w[i]$ with e itself; e.g., we talk about the source and target vertices of $w[i]$ to mean those of e. For $1 \leq i \leq \ell$ we call $w[i]$ a *first-visited* edge occurrence if it is the first occurrence of some edge e, i.e., if there is no $j < i$ such that $w[j] = w[i]$. Otherwise, $w[i]$ is a *revisited* edge. Of course, each edge that occurs in w is first-visited at exactly one position.

The *source vertex* s of w is that of $w[1]$, and its *target vertex* t is that of $w[\ell]$: we then call w an *st-walk*. The *length* $|w|$ of w is ℓ. The *set of vertices used by* w, denoted V_w, is simply the set of vertices that occur in w, i.e., occur in some edge of w; in particular the source and target vertices of w are in V_w. The *set of edges used by* w is $E_w := \{w[i] \mid 1 \leq i < \ell\}$. The *subgraph spanned by* w is $G_w = (V_w, E_w)$. Note that G_w is *not* the subgraph induced by V_w, as it only contains the edges that actually occur on the walk. A *subwalk* of w is a contiguous subsequence of w. We denote subwalks of w with the slicing convention, i.e., $w[: i] = w[1] \dots w[i]$ and $w[i : j] = w[i] \dots w[j]$. Note that the right endpoint is included so that, e.g., $w[i : j]$ is empty precisely when $j < i$, and $w[: i]$ is empty precisely when $i \leq 0$. When u and v are walks and when the target vertex of u is the source vertex of v, we write uv to mean the walk obtained by concatenating u and v: it admits u and v as subwalks, and of course $|uv| = |u| + |v|$.

For convenience, throughout the paper we denote by $G_{w,i}$ the graph $G_{w[:i]}$ spanned by the subwalk $w[: i]$ (up to i included); by convention $G_{w,0}$ is always the graph with the single vertex s and no edges.

Strongly connected components. A *strongly connected component* (SCC) of a graph G is a maximal set of vertices C of G such that, for any two distinct vertices $u, v \in C$, there is a directed path from u to v in G. As usual, belonging to the same SCC is an equivalence relation, so that SCCs form a partition of the vertices of G. We say that an SCC C of G is *non-trivial* if the subgraph of G induced by C contains at least one edge. This is the case if and only if C contains more than one vertex, or contains a single vertex on which there is a self-loop.

EWM and solutions. We study the *Edge-Minimum Walk of Modular Length* problem (EWM), as defined in the introduction: given a directed graph $G = (V, E)$, terminals s,t, and non-negative integers $r < q$, we wish to compute a subset E' of E such that (V, E') has an st-walk of $r \bmod q$ and E' is edge-minimum (i.e., the number of edges of G' is minimum). This phrasing is somewhat different from the one given in the introduction, in which we wanted to compute an edge-minimum walk. However, we can easily compute an edge-minimum walk from E' by a product construction, in time $O(|E'| \times q)$.

In the sequel, we only consider the computation of edge-minimum subsets of edges (rather than walks). When fixing inputs $G = (V, E)$, s, t, q, and r, we talk about a *candidate solution* to mean a subset E' of E such that (V, E') has an st-walk of length $r \bmod q$ but E' is not necessarily edge-minimum, and of an *optimal solution* to mean a candidate solution which is additionally edge-minimum.

Cutwidth. The *cutwidth* is a structural parameter of graphs. We show that there always exists an optimal solution to EWM with bounded cutwidth, which suffices to ensure that such solutions can be found with an efficient algorithm. We first recall the definition of cutwidth for undirected graphs. Let $G = (V, E)$ be an undirected graph. An *ordering* of G is a total order over the vertex set V of G. A *cut* (V_-, V_+) of G that respects the ordering $<$ is a partition $V_- \uplus V_+ = V$ such that $v_- < v_+$ for all $(v_-, v_+) \in V_- \times V_+$. We say that an edge e of G *crosses the cut* if it has one endpoint in V_- and one in V_+, i.e., $e \cap V_-$ and $e \cap V_+$ are both nonempty. Note that self-loops never cross cuts. The *cutwidth* of a cut (V_-, V_+) that respects $<$ is the number of edges that cross this cut, and the *cutwidth* of $<$ is the maximum cutwidth of a cut that respects $<$. The *cutwidth* of G is then the minimum cutwidth of an ordering of V.

We now define cutwidth for directed graphs. Let $G = (V, E)$ be a directed graph. Its *underlying undirected graph* is $G' = (V, E')$ where $E' = \{\{u, v\} \mid (u, v) \in E, u \neq v\}$: note that self-loops are not reflected in G'. We then define the *cutwidth* of G to be that of G'. We underscore that our definition of cutwidth for directed graphs is always the undirected cutwidth: we do not consider the directed cutwidth in this paper. We then define the *cutwidth* of a walk w of G as the cutwidth of the directed graph G_w.

4 Segment Decomposition of a Walk

Our proof of our main theorem (Theorem 1.1) consists of three steps, spanning this section and the next two sections. First, in this section we introduce the notion of the *segment decomposition* of a walk, and we show that any walk w can be transformed into a walk w' that satisfies the same modularity conditions (i.e., has the same remainder modulo q), uses a subset of the edges (i.e., $E_{w'} \subseteq E_w$), and has a segment decomposition with $O(\log q)$ segments. Second, in the next section, we will show how the cutwidth of the graph $G_{w'}$ spanned by w' can be bounded linearly in this number of segments, implying that optimal solutions to EWM have bounded cutwidth. Third, in Section 6, we show that this cutwidth bound makes it possible to solve EWM in polynomial time.

Segment decomposition. Let w be a walk. Its *segment decomposition* is a sequence of walks $s_1, \ldots, s_\xi$ such that $w = s_1 \cdots s_\xi$; the number ξ is the *number of segments* of w. The segment decomposition is defined by processing the walk w from left to right as follows. Initially, the segment decomposition is the empty sequence, and the entire walk w remains to be decomposed. At any point of the decomposition, we have already computed the first σ segments $s_1, \ldots, s_\sigma$ for some number $\sigma \geq 0$ of segments, and can write $w = s_1 \cdots s_\sigma w'$. If w' is non-empty, we compute the next segment as a prefix of w'.

Informally, we read w' and terminate the segment whenever we have a first-visited edge $w'[j] = (u, v)$ followed (not necessarily immediately) by an edge $w'[k] = (x, y)$ where y can be reached by a path from u using only edges of w up to $w'[j]$ excluded. The intuition of segments is that they capture a moment where the walk revisits a vertex y from a vertex u via a path that starts with an edge $e = (u, v)$ which had not been previously visited in w, even while there was already a path from u to y in the graph spanned by the walk before e was visited. Formally, we will look for the *segment end* inside the suffix w' of w, according to the following definition:

▶ **Definition 4.1** (Segment end, Segment detour). *Having fixed $w = s_1 \cdots s_\sigma w'$, let $\lambda_\sigma = |s_1 \cdots s_\sigma| = |w| - |w'|$ be the total length of the already-computed segments. For $\lambda_\sigma < k \leq |w|$, we say that segment $s_{\sigma+1}$* ends at *k if k is minimum such that the following hold:*

- *There is $\lambda_\sigma < j \leq k$ such that $w[j]$ is first-visited in w; we write $w[j]$ as (u, v);*
- *Writing the edge $w[k]$ as (x, y), then there is a path from u to y in the graph $G_{w,j-1}$ which contains the edges of the walk w up to $w[j]$ excluded. Note that, in particular, if $u = y$ then this requirement is always satisfied using the empty path from u to y.*

We stress that the path from u to y required by the second item of the definition is a path in the graph spanned *by a certain prefix of w; it may be the case that the path does not occur as a subwalk of w. As for the subwalk $w[j : k]$ from u to y in w, which is a subwalk of w', we call it the* segment detour *of $s_{\sigma+1}$, denoted $\det_{\sigma+1}$. Note that, having chosen the minimal k, there could be multiple valid choices for j, and we pick one arbitrarily.*

Given the definition of a segment end above, the next segment of the decomposition after s_σ is simply $s_{\sigma+1} := w[\lambda_\sigma + 1 : k]$. This means that we terminate the $(\sigma+1)$-th segment right after the edge $w[k]$, and continue the decomposition if $s_1 \cdots s_{\sigma+1}$ does not yet cover the entire walk w. If there is no segment end at any k, then we take all the rest of the walk w' to be the last segment of the decomposition and we finish. Note that for this reason, unlike other segments, the last segment of the decomposition may not feature a segment detour.

Let us formalize which paths are known to exist at the moment when a segment ends:

▶ **Lemma 4.2.** *Let w be a walk and let $w[i:k] = s_\sigma$ be the σ-th segment of the walk. Let y be the ending vertex of $w[i:k]$, and assume that $k < |w|$, i.e., the segment finishes before the end of the walk. Let $w[j] = (u, v)$ in $w[i:k]$ be the first edge of the segment detour $\det_\sigma$. There is:*

- *the segment detour $\det_\sigma$ from u to y in $G_{w,k}$*
- *a path p_σ from u to y in $G_{w,j-1}$.*
- *a path $\overline{p_\sigma}$ from y to u in $G_{w,j-1}$.*

Notice that we may have $y = u$, and then p_σ and $\overline{p_\sigma}$ may be empty; but $\det_\sigma$ is never empty because it contains at least $w[j]$.

In the sequel for each $1 \leq \sigma < \xi$ we denote by p_σ and $\overline{p_\sigma}$ a pair of paths obtained by applying Lemma 4.2 (if there are multiple valid choices for p_σ and $\overline{p_\sigma}$, we choose arbitrarily).

Achievable differences. For a walk w and segment σ, we denote by $\Delta_q(w, \sigma)$ the value $(|\det_\sigma| - |p_\sigma|) \bmod q$ and call this the[3] *difference achievable by σ*. We often drop the subscript q when clear from context. Intuitively, we know that we can modify the walk w to replace the subwalk $\det_\sigma$ from u to y by the existing path p_σ that goes from u to y, and this change subtracts $\Delta_q(w, \sigma)$ from the length of the walk modulo q; we will make this formal in the sequel. It should also be intuitively clear that achievable differences can be assumed to be non-zero in an edge-minimum walk: intuitively, an achievable difference of 0 means that we can replace the detour $\det_\sigma$ by the existing path p_σ that has the same endpoints, without changing the length of the walk modulo q. We know that this change does not harm the edge-minimality of w because p_σ consists of already visited edges. What is more, having too many segments with the same achievable difference is useless, because we can intuitively replace $\det_\sigma$ by p_σ and compensate the effect of this change by doing similar substitutions in earlier segments. Then, thanks to Lagrange's theorem on the additive group $\mathbb{Z}/q\mathbb{Z}$, the number of segments can in fact be bounded by $1 + \log_2 q$. We formalize this in the following lemma, which is the main result of this section:

▶ **Lemma 4.3** (Segment Decomposition Lemma)**.** *For every st-walk w with length $r \bmod q$, there is another st-walk of length $r \bmod q$ using only edges from w whose number of segments is at most $1 + \log_2 q$.*

The rest of this section is devoted to proving the Segment Decomposition Lemma; we will then use it in the next section to bound the cutwidth of some edge-minimum solution.

Recall that a *preorder* over a set is a binary relation that is both reflexive and transitive (this is a weaker requirement than non-strict partial orders which are additionally required to be antisymmetric). To prove the Segment Decomposition Lemma (Lemma 4.3), we define

[3] Note that there could be several choices for the difference achievable by σ depending on the arbitrary choices made earlier, e.g., depending on the first edge j for the segment detour, and on the choice of paths p_σ and $\overline{p_\sigma}$; nevertheless, we fix one single value $\Delta_q(w, \sigma)$ according to the choices made.

a preorder over the edge-minimum walks that are using a specific set of edges. Intuitively, the preorder favors walks which do their first visit of edges as late as possible relative to the end of the walk. More precisely, the preorder criterion asks us to minimize the number of remaining steps of the walk at the moment where the last first-visited edge is visited; then break ties by minimizing the number of remaining edges at the moment where the second-to-last first-visited edge is visited; and so on. This is designed to ensure that replacing a detour $\det_\sigma$ by p_σ improves the walk according to the preorder.

Let us fix from now on the graph $G = (V, E)$, the source s and target t, and the integers r and q. For any subset of edges $E' \subseteq E$, we define an *E'-walk* to mean an st-walk of length r mod q in G which uses precisely the edges of E'. Let us then define the order:

▶ **Definition 4.4.** *Let $E' \subseteq E$ be a subset of edges and let $m := |E'|$. Let w be an E'-walk, and let $i_1, \ldots, i_m$ be the indexes of the first-visited edges in ascending order, i.e., $w[i_1], \ldots, w[i_m]$ are the first-visited edges of w and $i_1 < \cdots < i_m$. The* first-visited timestamp *of w is then the m-tuple $(|w| - i_m, \ldots, |w| - i_1)$.*

We define as follows a preorder relationship $\trianglelefteq$ called the timestamp preorder *on E'-walks. Let w and w' be two E'-walks, and let t_w and t'_w be their first-visited timestamps, respectively. Then we have $w \trianglelefteq w'$ if we have $t_w \leq t'_w$ in lexicographic ordering.*

In other words, the timestamp preorder compares two walks by the number of remaining edges to traverse when visiting the last first-visited edge, then the second-to-last, and so on. Note that the timestamp preorder is not antisymmetric because two different walks on the same set of edges may happen to have the same first-visited timestamp (i.e., they visit first-visited edges at the same positions from the end, even though the identity of these edges and the revisits may be different). We then define:

▶ **Definition 4.5.** *If w is an E'-walk, we say w is* timestamp-minimum *if, for every E'-walk w', we have $w \trianglelefteq w'$.*

Note that, whenever there is an E'-walk, then there is a timestamp-minimum E'-walk, but it is not necessarily unique because the timestamp preorder is not antisymmetric. We are now ready to conclude the section by proving the Segment Decomposition Lemma (Lemma 4.3):

Proof sketch. We show that a timestamp-minimal walk has at most $1 + \log_2 q$ segments. We consider the successive subgroups of $\mathbb{Z}/q\mathbb{Z}$ generated by the achievable differences of the successive segments, and show that these must be a strictly increasing subsequence, so that the bound follows by Lagrange's theorem. To do this, we show as an intermediate claim that walks can be modified to augment their length by any combination of achievable differences of previous segments, while still using the same edges. Thanks to this claim, assuming by contradiction that there is a segment s_σ whose achievable difference is already achievable using the preceding segments, then we rewrite the segment to replace its detour $\det_\sigma$ by the path p_σ from Lemma 4.2, and use the claim to modify the walk and fix its length. We then show that this modification yields a walk which is smaller in the timestamp preorder, contradicting the minimality of w. See the full version [3] for the full proof. ◀

5 Number of Segments and Cutwidth Bounds

In this section, we continue our proof of Theorem 1.1 by showing that, for any walk w, the number of segments in the decomposition of the previous section gives a bound on the cutwidth of G_w up to a constant factor. This result is completely independent from the definition of the EWM problem. Formally, we show:

▶ **Proposition 5.1** (Segment Cutwidth Bound). *For any walk w, letting ξ be the number of segments in its segment decomposition, the cutwidth of G_w is at most 3ξ.*

We remark that there is no converse of this result: a walk formed of a succession of cycles connected by single edges will have an unbounded number of segments but has cutwidth 2. Combining Proposition 5.1 with Lemma 4.3 establishes the following:

▶ **Corollary 5.2.** *For every st-walk w' with length $r \bmod q$, there is another st-walk w of length $r \bmod q$ using only edges from w' such that the cutwidth of G_w is at most $3 + 3\log_2 q$.*

This implies the following bound on the cutwidth of optimal solutions to the EWM problem, on which our algorithm relies:

▶ **Corollary 5.3.** *For any graph $G = (V, E)$, terminals s,t, and non-negative integers $r < q$, every optimal solution to the EWM problem on G, s, t, r, and q has cutwidth at most $3 + 3\log_2 q$.*

To prove Proposition 5.1, we need several intermediate steps. First, we define the notion of *chunk* of a walk, which is a contiguous sequence of first-visited edges whose intermediate vertices are also first-visited. Second, we define the ordering $\prec$ on the vertices of G_w along which the cutwidth bound will be shown: this order is defined using the notion of chunks, intuitively because all vertices of a chunk will be ordered relative to the already-visited vertices that are the endpoints of the chunk (also distinguishing special cases like *cycle chunks* and *tadpole chunks*). Third, we show that the cutwidth along the order $\prec$ is bounded by 3ξ, by showing for each segment that the number of times its first-visited edges can cross the cut is at most 3. Throughout this section, we fix an arbitrary walk w in a graph G, and call s and t its source and target vertices.

Chunks. We define first-visited vertices similarly to first-visited edges: for every vertex $v \in V_w$ occurring in w, we say v is *first-visited at $w[i]$* if $w[i]$ is the first edge in which v occurs. Note that a vertex v which is first-visited at $w[i]$ always occurs as the target vertex of $w[i]$, except in the specific case of the first edge $w[1]$ where both the source s and the target vertex are first-visited at $w[1]$. Of course, every vertex of V_w is first-visited at exactly one position. Further, if a vertex v is the target vertex of an edge $w[i]$ and is first-visited at $w[i]$, then both $w[i]$ and $w[i+1]$ (if it exists) must be first-visited edges.

We can now define *chunks*:

▶ **Definition 5.4.** *A* chunk *of w is a maximal subwalk $w[i : j]$ of w where all edges are first-visited, and where, for every $i \leq k < j$ (if any), the target vertex of $w[k]$ is first-visited at $w[k]$.*

In other words, a chunk is a maximal sequence of one or more consecutive first-visited edges such that the first and last vertex are not first-visited (unless they are extremities of the whole walk) but all intermediate vertices are first-visited. A chunk may consist of a single first-visited edge $w[i]$ between two already-visited vertices, in which case there are no intermediate vertices.

The *first* and *last* vertices of chunk $w[i : j]$ are the source vertex of $w[i]$, and the target vertex of $w[j]$, respectively. Note that two successive chunks in the walk need not be separated by revisited edges and can simply be separated by a revisited vertex: for instance, for the length-2 walk $(s, s), (s, t)$, all edges are first-visited, but there are two chunks of length 1 which are separated by the revisit of the vertex s.

It will be useful to distinguish two special kinds of chunks. First, *tadpole* chunks, which loop back on an intermediate vertex of the chunk:

▶ **Definition 5.5.** *A chunk is a* tadpole *if it consists of at least two edges and if, letting $u_1, \ldots, u_\ell$ be the successive vertices that it visits, with u_1 its first vertex and u_ℓ its last vertex, we have $u_\ell = u_{\ell'}$ for some $1 < \ell' < \ell$.*

In other words, a *tadpole* is a chunk that consists of a path (containing at least one edge), followed by a cycle (possibly a self-loop).

Second, *cycle* chunks, which loop back on the vertex from which they started:

▶ **Definition 5.6.** *A chunk is a* cycle *if its first vertex and last vertex are identical.*

Note that these two cases are mutually exclusive. A chunk which is neither a tadpole nor a cycle, and thus is a simple path, is a *normal chunk*.

The notion of chunk must be distinguished from the notion of segments used in the segment decomposition of the previous section, but we can notice the following connections between the two notions:

- Segment ends never happen within a chunk: indeed, the intermediate vertices y reached within a chunk are first-visited by definition, so there is no way to reach them except by the edge that precedes them, and thus no earlier path can reach y in the walk.
- At the end of a tadpole chunk or cycle chunk, the current segment always ends. Indeed, the last edge $e' = (x, y)$ of the chunk reaches a vertex y which already has an outgoing edge in the chunk: this edge is a first-visited edge $e = (y, v)$, and the empty path from y to itself allows us to finish the segment at the end of the chunk, i.e., just after e'.

In summary, chunks are contiguous subsequences of the walk that never straddle segment boundaries, and the end of a tadpole chunk or cycle chunk always triggers the end of a segment. Also note that, by definition, chunks form a partition of the first-visited edges. This implies that, except possibly for the last segment, every segment must contain at least one chunk, because they contain at least one first-visited edge.

Defining the ordering. Having defined the notion of chunks of the walk w, we now define the ordering $\prec$ along which we will show that the cutwidth is bounded. This definition only depends on chunks; it does not depend on the segment decomposition. We see the total order $\prec$ as a sequence of vertices. Initially, the order is the empty order on the single vertex s. Then, for every chunk $w[i : j]$ successively, we consider the (possibly empty) sequence σ of the intermediate vertices of $w[i : j]$. Recall that the first vertex of $w[i : j]$ is already in the domain of $\prec$: either it is s (for the first chunk), or it is a vertex which is already visited. Then there are four cases:

- Tadpole: If $w[i : j]$ is a tadpole, then we insert all its intermediate vertices at the end of the current ordering $\prec$, in the order in which they were first-visited.
- Cycle: Otherwise, if $w[i : j]$ is a cycle, then, letting v be its first and last vertex, we know that v already occurs in $\prec$, and we insert all its intermediate vertices right after the vertex v, in the order in which they were visited.
- Normal: We consider two subcases:
  - First, suppose that the last vertex t of $w[i : j]$ is first-visited at $w[j]$. This is only possible with $j = |w|$, so that $w[i : j]$ is the last chunk. Then we do the same as in the tadpole case: insert the intermediate vertices and t at the end of the current ordering $\prec$, in the order in which they were first-visited.
  - Otherwise, $w[i : j]$ is a chunk whose last vertex v is first-visited at $w[k]$ with $k < i$ so v already occurs in $\prec$, and as we explained the first vertex u of the chunk also already occurs in $\prec$. Further, we have $v \neq u$ because the chunk is not a cycle. Then, we insert

the intermediate vertices so that the vertices along the chunk are ordered in a monotone fashion in the ordering. In other words, let $x_1, \ldots, x_\ell$ be the successive intermediate vertices of the chunk, so that its edges are $(u, x_1), (x_1, x_2), \ldots, (x_{\ell-1}, x_\ell), (x_\ell, v)$. If $u \prec v$ then we insert $x_1, \ldots, x_\ell$ in order between u and v, and if $v \prec u$ we insert $x_\ell, \ldots, x_1$ in order between v and u. The order $\prec$ between the newly inserted elements and the existing elements between u and v is arbitrary, for instance we can arbitrarily say that we insert the new vertices just after the smallest of u and v in $\prec$.

Note that, in all cases above, the (intermediate) vertices of a chunk are always ordered as a monotone sequence.

SCCs of the segment decomposition. We have defined, from our walk w, the order $\prec$ along which we will bound the cutwidth. To show the cutwidth bound, we establish two results describing the SCCs of the graph generated by the walk and its close relationship with our ordering $\prec$:

▶ **Lemma 5.7.** *At each step $0 \leq i \leq |w|$, the successive SCCs $C^i_1, \ldots, C^i_{\kappa_i}$ of $G_{w,i}$ are linearly ordered by the reachability relationship (i.e., every vertex in C^i_b is reachable from every vertex in C^i_a iff $a < b$), and the target vertex of the last edge $w[i]$ is in the last SCC $C^i_{\kappa_i}$ of $G_{w,i}$.*

▶ **Lemma 5.8.** *For every prefix $w[:i]$ of the walk such that a chunk of w ends at $w[i]$, the ordering $\prec$ induced by $w[:i]$ is consistent with the topological order of the SCCs of $w[:i]$.*

Bounding the cutwidth from the number of segments. We can now turn back to the segment decomposition introduced in the previous section for the walk w, and show how the number of segments of w can be used as a bound on the cutwidth of the graph G_w spanned by w, following the order $\prec$ that we defined. For this, we will consider an arbitrary cut $V_- \uplus V_+$ of V_w following $\prec$, and count how many edges of w cross the cut. As each edge is first-visited once, and first-visited in exactly one segment, it suffices to bound how many edges each segment contributes to the cut, and to count only the first-visited edges of each segment. We want to show:

▶ **Lemma 5.9.** *For each segment $w[i:j]$ of the walk w, the number of first-visited edges in $w[i:j]$ that cross the cut is at most 3.*

This immediately implies the Segment Cutwidth Bound (Proposition 5.1) stated at the beginning of the section. This bound of 3 edges crossing the cut can be proved by a case analysis summarized in the following lemmas:

▶ **Lemma 5.10.** *Let $w[i:j]$ be a segment of w and assume that the last chunk of $w[i:j]$ is a cycle $w[k:j]$. Then $w[k:j]$ crosses the cut at most twice. Further, if it does cross the cut twice then it must be the case that the first and last vertex of $w[k:j]$ is in V_- and that $w[k:j]$ contains some intermediate vertex in V_+.*

▶ **Lemma 5.11.** *Let $w[i:j]$ be a segment of w and assume that the last chunk of $w[i:j]$ is a tadpole $w[k:j]$. Then $w[k:j]$ crosses the cut at most twice. Further, if it crosses the cut twice then it has an intermediate vertex in V_- and an intermediate vertex in V_+.*

▶ **Lemma 5.12.** *Let $w[i:j]$ be a normal chunk of w, then it crosses the cut at most once.*

▶ **Lemma 5.13.** *Let $w[i:j]$ be a normal chunk of w which crosses the cut forwards, i.e., the starting vertex of the chunk is in V_- and its ending vertex is in V_+. Then the segment containing $w[i:j]$ ends at j.*

All that remains is to bound the contribution to the cut of the normal chunks that cross the cut backwards. To this end, let us distinguish the last chunk of a segment which we call the *final* chunk, and the remaining chunks in that segment which we call *non-final* chunks. Non-final chunks must be normal (because the segment ends right after cycle chunks or tadpole chunks), and they cannot cross the cut forward by the previous lemma.

We are ready to conclude the proof of Lemma 5.9:

Proof sketch. We consider a segment and consider the SCC decomposition of the graph of the walk when the segment starts, using Lemma 5.7. By Lemma 5.8 the segment begins in the rightmost SCC according to the order. Now the intuition is the following: if all chunks but the final one stay on the same side of the cut, then only the final chunk can cross the cut and the preceding lemmas (Lemma 5.10, Lemma 5.11 and Lemma 5.12) show that the bound is satisfied.

Otherwise, there is a first non-final chunk which ends on some SCC C_ϕ that contains nodes to the left of the cut and this chunk thus potentially crosses the cut (backwards, by Lemma 5.13). If the next chunk is final, the bound is satisfied, so we may assume the next chunk is not final. This next chunk may also cross the cut, but in any case it has for target an SCC C_ψ which is further left than C_ϕ, a property we exploit to show that the walk cannot return to the right of the cut without terminating the segment. This implies that the cut is crossed at most 4 times in total: once for each of the two non-final chunks considered, and potentially twice for the final chunk. We lower the bound to 3 by showing that in fact the final chunk can only cross the cut once if it is preceded by two chunks that already crossed the cut. See the full version [3] for the full proof. ◀

6 Computing an Edge-Minimum Walk of Bounded Cutwidth

Up to now, we have shown Corollary 5.3: optimal solutions to the EWM problem have cutwidth at most $3 + 3\log_2 q$. In this section, we conclude the proof of Theorem 1.1 by showing that optimal solutions of bounded cutwidth can be efficiently found.

At a high level, our algorithm proceeds by searching for a shortest path in a graph of configurations. The approach is similar to the approaches in [17] and [18]. The intuitive principle of our algorithm is the following. As we follow a path in the graph of configurations and move from one configuration to another, we choose which edges of the original graph to keep in the solution. To be more precise, the path will start on the empty configuration which denotes that no edges have been selected, and will iteratively add vertices and some of their incident edges to the solution.

To make sure that the selected edges do form a solution, we could try to record in the configurations the exact set of edges kept in the solution so far. However, the space of configurations would then be exponential. To avoid this, a configuration does not really store the entire set of chosen edges: instead it concisely represents the possible lengths modulo q of walks between a small subset of vertices of G, whose size is bounded as a function of the cutwidth.

The distance in the configuration graph from the initial configuration to any configuration Ξ will reflect the smallest cardinality of a set of edges that achieve the set of walks witnessed by Ξ. The algorithm therefore looks for a shortest path in the configuration graph from the initial configuration to any configuration which witnesses the st-walk of length r mod q required by the EWM problem.

Formally, fix the input graph $G = (V, E)$, the source $s \in V$ and the target $t \in V$, and the modularity requirement $r \bmod q$. Let $\omega \in \mathbb{N}$ be a *domain size bound*, which is related to the cutwidth, but will be precisely defined later. Let us define the notion of configurations formally:

▶ **Definition 6.1** (Configuration). *A V, ω, q-*configuration *$\Xi = (D, \rho)$ is a subset D of at most ω vertices of V, called the* domain *of the configuration, together with a function ρ mapping each ordered pair (u, v) for $u, v \in D$ to a subset of $\{0, \ldots, q-1\}$. Having fixed the vertex set V of the input graph G, we denote by $\Phi_{\omega,q}$ the set of all possible V, ω, q-configurations, and omit the subscript when clear from context.*

To compute transitions in the configuration graph, our algorithm will need to compute the *closure* of a configuration (D, ρ). The point of the closure is to ensure that, whenever we can achieve a walk from $u \in D$ to $v \in D$ via intermediate vertices of D and using remainders given by ρ, then that walk can be witnessed directly by a value in $\rho(u, v)$. Formally:

▶ **Definition 6.2** (Closure). *Given a configuration $\Xi = (D, \rho)$, an* internal walk *of Ξ is a sequence of vertices of D together with a choice of remainder for each step. Formally, it is a sequence $v_1, \ldots, v_\ell \in D$ and a choice of remainders $0 \leq r_1, \ldots, r_{\ell-1} < q$ such that we have $r_i \in \rho(v_i, v_{i+1})$ for each $1 \leq i < \ell$. The* total length *of the internal walk is the sum of the remainders modulo q, namely, $\sum_{1 \leq i < \ell} r_i \bmod q$.*

The closure *of (D, ρ) is the configuration (D, ρ') where, for each $u, v \in D$, for each $r' \in \{0, \ldots, q-1\}$, we have $r' \in \rho'(u, v)$ iff there is an internal walk of total length r' from u to v in (D, ρ). Note that we always have $\rho'(u, v) \supseteq \rho(u, v)$ because for each $r' \in \rho(u, v)$ we can take the single-edge internal walk u, v with remainder r'.*

We can easily compute the closure of a configuration in polynomial time in q and ω, for instance using a product construction. Specifically, create a graph on vertices $D \times \{0, \ldots, q-1\}$, then add the following edges: for each $u, v \in D$ and each $r' \in \rho(u, v)$, for each $i \in \{0, \ldots, q-1\}$ create an edge from (u, i) to $(v, i + r' \bmod q)$. Then compute the transitive closure and define $\rho'(u, v)$ to be the set of r' such that $(u, 0)$ has a path to (v, r').

We now define the graph over configurations, which we call $\Gamma_{\omega,q}$ and which also depends on the directed graph G given as input to EWM. We omit the subscripts when clear from context.

▶ **Definition 6.3.** *The* graph of configurations $\Gamma_{\omega,q}$ *is a weighted and labeled directed graph: each edge carries an integer cost and is labeled by a subset of edges of the original graph G. The vertex set of Γ is the set Φ of all V, ω, q-configurations. To define the edges, let us choose any configuration $\Xi \in \Phi$ and define the outgoing edges of Ξ. These edges are of two kinds:*

- Forget*: from $\Xi = (D, \rho)$ with D nonempty, for each $v \in D$, letting $D' := D \setminus \{v\}$ be the new domain, we have an edge leaving Ξ which has cost 0, is labeled with the empty set of edges, and leads to $(D', \rho_{|D'})$ where $\rho_{|D'}$ is the restriction of ρ to D'*
- Introduce*: from $\Xi = (D, \rho)$ with $|D| < \omega$, for each $v \in V \setminus D$, let $D' := D \uplus \{v\}$ be the new domain, and let $E_{v,D'}$ be the set of edges of G which are of the form (v, v') or (v', v) with $v' \in D'$; we also add to $E_{v,D'}$ the self-loop edge on v if it exists. Then for each $E' \subseteq E_{v,D'}$, we have an edge leaving Ξ which has cost $|E'|$, is labeled by E', and leads to the closure of the configuration (D', ρ') with ρ' intuitively defined from ρ by adding the edges of E', formally:*
  - *For all $u, u' \in D \cup \{v\}$, we initialize $\rho'(u, u') := \emptyset$.*
  - *For each $(u, u') \in D \times D$, we set $\rho'(u, u') := \rho(u, u')$.*
  - *For each edge $e \in E_{v,D'}$, we set $\rho'(e) := \{1\}$. Note that by definition of $E_{v,D'}$ this case is disjoint from the previous one.*

Note that we may have several Introduce edges with the same source and target configurations but labeled with different sets of edges and with different costs; in this case all of these edges exist in Γ *as defined above, although we could equivalently have decided to keep only one of these edges among those with minimum cost.*

We have now defined the graph Γ of configurations. We will look for shortest paths in this graph from the *initial configuration* to a *final configuration*, namely:

▶ **Definition 6.4.** *The* initial configuration *is simply the configuration with empty domain.*

A configuration is final *if it contains* s *and* t *and features a walk with the requisite remainder, i.e., the configuration* (D, ρ) *is final if* $s, t \in D$ *and* $r \in \rho(s, t)$*.*

We can now define our algorithm to solve the EWM problem, which we call the *EWM algorithm*. We first set the value ω, following the cutwidth bound, to be: $\omega := 6 + 3\log_2 q$, i.e., three more than the cutwidth bound that follows from Corollary 5.3. (Intuitively, we add 2 to make sure that the sources s and t can always be part of the domain of configurations, and we add 1 extra to make sure that we can always perform Introduce steps before Forget steps.) The EWM algorithm then builds explicitly the graph $\Gamma_{\omega,q}$ and computes a shortest path π in Γ from the initial configuration to a final configuration. Once such a shortest path π is found[4], then the algorithm returns the subgraph of G formed of the edges of G obtained as the union of all edge labels in π.

Proving that the EWM algorithm correctly solves the EWM problem is relatively straightforward: we show that whenever the algorithm return a subgraph, this subgraph is indeed a candidate solution. Conversely, we show that for every optimal solution E' there is a path from in the configuration graph from some initial configuration to some final configuration, such that the union of labels along the path is E' without duplicates (hence the path has cost $|E'|$), which guarantees that the algorithm returns the optimal solution. Those properties can be proved by establishing the following invariant:

▷ Claim 6.5. Let $\pi = \Xi_1, \ldots, \Xi_\ell$ be a path in Γ from the initial configuration Ξ_1. Let $G_1 = (V, E_1)$, …, $G_\ell = (V, E_\ell)$ be the sequence of subgraphs of G defined in the following way: we have $E_1 = \emptyset$, and for each $1 < i \leq \ell$ we set $E_i = E_{i-1} \cup E'_i$ where E'_i is the edge set that labels the edge of Γ used to go from Ξ_{i-1} to Ξ_i in Γ. Then the following is true: for any $1 \leq i \leq \ell$, writing $\Xi_i = (D_i, \rho_i)$, for any $u_1, u_2 \in D_i$, for any $r' \in \{0, \ldots, q-1\}$, we have $r' \in \rho_i(u_1, u_2)$ iff there is a u_1u_2-walk in G_i whose length modulo q is r'.

Note that, in the definition of the graphs G_i, we may add the same edge of G multiple times because it occurs in the label of several edges of π; this can happen even if π is a simple path in Γ. As it turns out, this never happens when π is a shortest path (because we are then, in effect, paying twice for the same edge); but the invariant of Claim 6.5 applies to paths π even if they do traverse edges labeled with non-disjoint edge sets.

Altogether, we have explained why the algorithm correctly solves the EWM problem, and the algorithm can be shown to have complexity $n^{\omega+1} \cdot 2^{O(\omega^2 q)}$. Instantiated with our choice of $\omega = 3 + 3\log_2 q$, we get a final complexity of $n^{O(\log q)} \cdot 2^{O(q \log^2 q)}$ for the algorithm. See the full version ([3]) for details. This concludes the proof of our main result (Theorem 1.1).

[4] If there is no path in Γ from an initial configuration to a final configuration, then we return $\emptyset$ to indicate that there is no subgraph of G with an st-walk of length r mod q. Note that this case can be excluded from the outset, simply by checking whether G contains an st-walk of length r mod q: this can be done, e.g., with the product construction.

7 Extension to Weighted Graphs

Having shown our main result (Theorem 1.1), in this section we start exploring variants and extensions of the EWM problem. We first study two extensions of EWM in this section. First, we study the addition of *costs* on edges, meaning that we look for a walk where the total cost is minimum (instead of the number of edges). Second, we study the addition of integer *lengths* on edges.

In this section and the next, the problems that we define and study are always posed on a directed graph G and ask about the existence of a subgraph of G satisfying certain properties and which is optimal according to some criterion (usually, being edge-minimum). We use the same terminology as before: a *candidate solution* is a subgraph which satisfies the properties but is not necessarily edge-minimum, and an *optimal solution* is a candidate solution which is additionally edge-minimum.

Costs on edges and vertices. We first study the extension of the EWM problem with costs on edges. Specifically, the *EWM problem with costs on edges* takes as input a directed graph $G = (V, E)$, a cost function γ giving to each edge $e \in E$ a cost $\gamma(e)$, a pair of a source $s \in V$ and target $t \in V$, and a modularity requirement r mod q for integers q and r. The output to the EWM problem with costs on edges is an st-walk of length r mod q such that the cost $\gamma(E_w) := \sum_{e \in E_w} \gamma(e)$ is minimum. We assume that all costs given by γ are nonnegative. Indeed, in the presence of negative weights, we lose the correspondence explained in Section 3: computing a minimum-weight *subgraph* that contains a walk reduces to the case of positive weights (all negative-weight edges will always be included in the optimal solution subgraph); but computing a minimum-weight *walk* is NP-hard even without modularity constraints:

▶ **Proposition 7.1.** *The following problem is NP-complete: given a directed graph $G = (V, E)$, a cost function $\gamma \colon E \mapsto \mathbb{Z}$, and source and target $s, t \in V$, compute a minimum-weight subset of edges $E' \subseteq E$ such that there is an st-walk using precisely the edges in E'.*

Our tractability result for EWM (Theorem 1.1) can be easily extended to solve the EWM problem with costs on edges. Specifically, the Segment Decomposition Lemma (Lemma 4.3) shows that there must be an optimal solution to EWM with costs on edges that obeys the bound on the number of segments, because the γ function on subset of edges is monotone. Then the Segment Cutwidth Bound (Proposition 5.1) applies as is, and the algorithm to find an optimal solution of bounded cutwidth from the previous section can be easily extended: simply modify the definition of the graph of configurations (Definition 6.3) so that the cost of an edge labeled with a set E' of edges of G is no longer the cardinality $|E'|$ of E but the total cost $\gamma(E')$. Other than that, the algorithm proceeds in the same way, and computes a shortest path which now reflects the subgraph of G satisfying the requirements which has minimum cost instead of minimum cardinality.

We also mention that the EWM problem can be defined to have costs (unit costs or not) on vertices instead of edges. In this case, the cost of a candidate solution is the number of different vertices traversed by the walk (or more generally their total cost). However, this problem is interreducible to the EWM problem with costs on edges:

▶ **Lemma 7.2.** *The EWM problem (with unit costs on edges) reduces to the EWM problem with unit costs on vertices, and the EWM problem with costs on edges reduces to the EWM problem with costs on vertices. Conversely, the EWM problem with unit costs on vertices reduces to the EWM problem with unit costs on edges, and the EWM problem with costs on vertices reduces to the EWM problem with costs on edges.*

Lengths on edges. Having studied the use of costs on edges to change the optimization criterion, we turn to a different problem variant where we annotate edges with integer lengths. Specifically, the *EWM problem with lengths* takes as input a directed graph $G = (V, E)$, a length function $\delta\colon E \to \mathbb{N}$, and a pair of a source $s \in V$ and target $t \in V$ and a modularity requirement $r \bmod q$ for integers q and r. The answer to the EWM problem with lengths is an st-walk w that traverses a minimum number of distinct edges and whose total length is $r \bmod q$, i.e., $\delta(w) := \sum_{1 \leq i \leq |w|} \delta(w[i])$ is $r \bmod q$. (Note that the length of an edge is summed as many times as the edge is traversed.)

The impact of allowing lengths is different depending on the regime. In the case where r and q are constant, or given in unary, then we can rewrite the graph in polynomial time to eliminate edge lengths. Specifically, letting m be the number of edges in the graph, we replace each edge e of length $\delta(e)$ by a path on $(m+1)q + (\delta(e) \bmod q)$ edges (where m is the number of edges in the graph). This ensures that a walk in the new graph has the same modularity as the corresponding walk in the original graph, and the cost of a candidate solution in the rewritten graph is $(m+1)qM + \epsilon$, where M is the number of original edges traversed and $\epsilon \leq mq$. This ensures that an optimal solution in the original graph indeed minimizes the number of edges taken from the original graph.

One different regime is when r and q are written in binary and do not necessarily have polynomial value. In this case, the EWM problem with lengths written in binary can be shown to be NP-hard by an easy reduction from Subset Sum. In fact, just the problem of deciding the existence of a walk of length $q \bmod r$ is NP-hard:

▶ **Proposition 7.3.** *The problem of deciding whether an st-walk of length $q \bmod r$ exists, where q, r and the lengths of δ are written in binary, is NP-hard.*

We do not know whether an analogous hardness result holds for the original EWM problem (without edge lengths) in the setting where q and r can be written in binary and do not necessarily have polynomial value. Indeed, in the proof above, we crucially use the edge lengths to concisely write the numbers given in the Subset Sum instance; writing them in unary as paths of the corresponding length will not give a polynomial-time reduction and indeed the Subset Sum problem can be solved in pseudo-polynomial time.

We last address the question of showing an upper bound on the complexity of EWM with lengths, by showing an NP upper bound. Of course the bound is phrased on the decision version of EWM with lengths, i.e., given an instance of EWM with lengths and a threshold k, we wish to decide whether there exists a candidate solution having at most k different edges. We have:

▶ **Proposition 7.4.** *The following problem is in NP: given a directed graph $G = (V, E)$, a function $\gamma\colon E \to \mathbb{N}$ that assigns lengths (written in unary) to each edge, integers q, r and k (written in binary), source and target $s, t \in V$, decide if there is a subset $E' \subseteq E$ of at most k edges that contains an st-walk of length $r \bmod q$ according to γ.*

8 Connections to DSN and SCSS

We conclude the paper by exploring how EWM relates to the Directed Steiner Network (DSN) problem mentioned in the introduction, and more specifically the *Strongly Connected Steiner Subgraph (SCSS)* problem. We first define the SCSS problem and show that it is subsumed by EWM, in the sense that a polynomial algorithm for EWM (with fixed q and r) gives a polynomial algorithm for SCSS with a constant number of terminals. Then, we introduce a problem generalizing both SCSS and DSN on the one hand, and EWM in the other: we study

how to find the smallest subgraph satisfying connectivity requirements on specified endpoint pairs with specified modularities. We show that we can generalize our results to this problem, and provide a polynomial algorithm for the setting when the number of endpoint pairs and the modularity requirements are constants.

Reducing SCSS to EWM. Let us recall the definition of the SCSS problem [17, 18] before explaining how it can be reduced to EWM. In the SCSS problem, the input simply consists of a directed graph $G = (V, E)$ with an input set $T \subseteq V$ of terminals, and we want to find the smallest subgraph $E' \subseteq E$ that is strongly connected and contains edges incident to each terminal in T. In other words, SCSS is a special case of DSN where the connectivity requirements on the vertices of T require that they are strongly connected. The SCSS problem is NP-hard in general (by an easy reduction from the directed Steiner tree problem) but it can be solved in polynomial time provided that the size of T is bounded by a constant, as shown in [17]. Thus, for $k \in \mathbb{N}$, we write k-SCSS to refer to the SCSS problem where the set T of terminals is required to contain at most k vertices. Following our previous terminology in the paper, when we talk of *candidate solutions* we mean a subgraph satisfying the requirements of a problem (e.g., for k-SCSS, a strongly connected subgraph containing the requisite terminals), and by *optimal solution* we mean a candidate solution which is also edge-minimum.

Let us show that the SCSS problem can be reduced to EWM, which implies that Theorem 1.1 gives an alternative proof of the results of [17] (with worse bounds). More precisely, we show:

▶ **Lemma 8.1.** *For any constant $k > 0$, we can compute constants q and r in $2^{O(k \log k)}$ such that there is a linear-time reduction from k-SCSS to EWM with the constant values q and r.*

The previous reduction shows that our EWM problem is intuitively at least as complicated as SCSS, given that the tractability of EWM for constant q implies that of SCSS for constant k. (This being said, we do not claim that the polynomial algorithm given by our results is as efficient as that of earlier works [17, 18].) Alternatively, instead of this black-box reduction, we can also adapt our techniques to recapture the tractability of SCSS by showing a bound on the cutwidth of optimal solutions, using the segment decomposition. We exemplify below how this is done, and will revisit this afterwards when extending EWM to support multiple paths. Recall the notion of timestamp-minimum walks (Definition 4.5), and let us show the following result; note that it applies to *edge-minimal* solutions (not just optimal solutions).

▶ **Lemma 8.2.** *Let $G = (V, E)$ and $T = \{v_1, \ldots, v_k\}$ be an SCSS instance. Every edge-minimal solution H to the instance has cutwidth at most $3k$.*

What is more, for any terminal v_i, considering all H-walks that start and conclude in v_i and visit all terminals from T, then any timestamp-minimum H-walk among those H-walks has at most k segments.

We remark that the cutwidth bound of $3k$ given by this result is slightly better than the bound of $6k$ shown in [18].

Defining Directed Steiner Network with Modularity constraints. The reduction from SCSS to EWM leads to the natural question of whether there could be a problem which subsumes SCSS and EWM, or even DSN and EWM; and a polynomial-time algorithm that subsumes the tractability of all these problems. We now introduce such a problem, dubbed *Directed Steiner Network with Modularity* (DSNM), and show a polynomial-time algorithm to solve it.

▶ **Definition 8.3.** *For $k > 0$, the* k-Directed Steiner Network with Modularity *problem (k-DSNM) takes as input a graph (V, E) and a* k-requirement specification $\mathcal{R}$ *consisting of k tuples (s_i, t_i, r_i, q_i) for $1 \leq i \leq k$ such that $s_i, t_i \in V$ and $q_i > 0$ and $0 \leq r_i < q_i$. Our goal is to compute an optimal solution, i.e., an edge-minimum subgraph H of G such that H contains a walk from s_i to t_i of length $r_i \bmod q_i$ for every $i \leq k$.*

Our last result in this paper is to show that the k-DSNM problem is in PTIME when k and the modulo values q_i are constants. This result subsumes Theorem 1.1 and the tractability of k-DSN for constant k shown in [17, 18].

▶ **Theorem 8.4.** *We can compute in $n^{O(k+\log q)} \cdot 2^{O(q(k+\log q)^2)}$ time an optimal solution to k-DSNM, where q denotes the least common multiple (LCM) of the q_i in the input k-requirement specification.*

Note that our exponents are given up to constant factors, so we do not claim to recover the same exponents as earlier works on k-DSN without modularity constraints (i.e., for $q = 1$).

The overall strategy to prove Theorem 8.4 is to follow the methodology used for EWM, adjusting the constructions presented so far in the paper. Deviating somewhat from EWM, but following prior work about DSN [17, 18], in our study of DSNM we will focus on the SCCs of an optimal solution and study them separately, instead of studying the entire solution graph. Our main objective is to bound the cutwidth of SCCs in an optimal solution as a function of q and of the number k of endpoint pairs. We can then easily deduce like in [18] that the cutwidth of optimal solutions is bounded as a function of q and k. To compute the solutions of bounded cutwidth, we then modify slightly the algorithm of Section 6.

To study the SCCs of optimal solutions, it will be useful to introduce an analogue of the SCSS problem featuring modularities. Specifically, we call k-SCSSM the problem that takes the same input as k-DSNM and returns a smallest *strongly connected* subgraph H of G such that H contains a walk from s_i to t_i of length $r_i \bmod q_i$ for every $i \leq k$. Note that, unlike the SCSS problem which was simply defined by a set of terminals, in k-SCSSM we specify a set of tuples connecting pairs of vertices, because we need to specify which remainder must be achieved by which endpoint pair. The difference with k-DSNM is only that the solution graph is required to be strongly connected. Our study of k-SCSSM is motivated by the fact that the SCCs of optimal solutions to k-DSNM are optimal solutions to instances of k-SCSSM, as was already known in the setting without modularities [17]. Namely:

▷ Claim 8.5. Let $G = (V, E)$ be a directed graph, and $\mathcal{R}$ be a k-requirement specification. In any optimal solution $H = (V, E')$ to k-DSNM on G for $\mathcal{R}$, for any SCC C of H, there exists a k-requirement specification $\mathcal{R}'$ such that C (as a set of edges) is an optimal solution to k-SCSSM on G for $\mathcal{R}'$.

We next bound the cutwidth of optimal solutions to the k-SCSSM problem specifically. Then we will deduce a cutwidth bound on optimal solutions to k-DSNM using a result of [18]. Finally, we explain how to design an algorithm to compute these solutions.

Bounding the cutwidth of solutions to k-SCSSM Let us start by showing that solutions to k-SCSSM have a small segment number and hence a small cutwidth:

▶ **Lemma 8.6.** *Let G be a graph and $\mathcal{R}$ be a k-requirement specification, denote by $q_1, \ldots, q_k$ the respective modularities of its tuples, and denote by q the least common multiple (LCM) of $q_1, \ldots, q_k$. Then every optimal solution of the k-SCSSM problem on G and $\mathcal{R}$ can be covered with a walk of at most $O(k + \log q)$ segments and therefore has cutwidth at most $O(k + \log q)$.*

Note that this claim only works for the k-SCSSM problem, not the k-DSNM problem, because it assumes the solution can be covered by a single walk which is not true in general for k-DSNM. Let us show the result:

Proof sketch. The proof of this result can be decomposed into three steps. In step 1, we define the notion of a *legal covering walk*, which is a walk that covers the edges of a solution in a prescribed order: first visit all terminals to witness that they are strongly connected in a subwalk w_0 (similarly to Lemma 8.2), then visit all paths with the prescribed modularities in a subwalk w'. In step 2, we carefully impose a variant of timestamp-minimality on legal covering walks towards bounding their number of segments. In step 3, we use Lemma 8.2 (on the first part of the legal covering walk) and an argument adapted from Lemma 4.3 (for the second part) to show that the number of segments is bounded. The point of splitting the walk in two is that w_0 visits enough edges to allow us to move to arbitrary endpoints and ensure strong connectedness: this allows us to replay arbitrary detours in any subwalk no matter where they occur, intuitively "pooling" the achievable differences $\Delta(w, \dots)$ between all the subwalks. See the full version [3] for the full proof. ◀

From k-SCSSM to k-DSNM. We have shown that optimal solutions to the k-SCSSM problem can be assumed to have bounded cutwidth. We now wish to show that the same is true for optimal solutions to the k-DSNM problem. This is simple, using Lemma 8.6 above along with two lemmas from [18]:

▶ **Lemma 8.7.** *Fix a graph G and a k-requirement specification $\mathcal{R}$, and let q be the LCM of the q_i in $\mathcal{R}$. Let H be an optimal solution to k-DSNM on G and $\mathcal{R}$. Then H has cutwidth at most $O(k + \log q)$.*

Computing optimal solutions. We have shown in Lemma 8.7 that optimal solutions to the k-DSNM problem have cutwidth $O(k+\log q)$. The only remaining thing to prove Theorem 8.4 is to adapt the dynamic algorithm from Section 6 to compute solutions to that problem instead of EWM. We explain in the full version [3] how this is done.

9 Future work

We now outline possible directions for future work in light of our results:

Improving our bounds. One natural direction for further research would be to improve the complexity bounds that we show. Our algorithm for EWM runs in time $n^{O(\log q)} \cdot 2^{O(q \log^2 q)}$. Can the factor of q in the exponent be improved, e.g., by getting an algorithm in time $n^{O(\log q)}$? Or, on the other hand, is the problem NP-hard? (We have only shown that the variant of EWM with edge lengths is NP-hard, in Proposition 7.1.)

Intermediate problems between shortest walk and edge-minimum walk. It could be interesting to search for walks that are minimized according to some criterion that is "intermediate" between shortest walk and edge-minimum walk, e.g., if the cost of each edge is expressed as a function of how many times it is traversed by the walk. Such a general framework would capture the two problem variants that we contrast in the introduction: shortest walks are the case where we are charged ℓ when traversing an edge ℓ times, and edge-minimum walks are the case where we are charged 1 when traversing an edge $\ell > 0$ times and charged 0 when we do not traverse it.

Edge-minimum walks satisfying other constraints. Last, one other problem of interest would be to investigate the complexity of finding edge-minimum subgraphs guaranteeing the existence of st-walks satisfying other properties. One very natural example is the following: Given a number ℓ, and a directed graph G with specified vertices s, t, find an edge-minimum st-walk of length exactly ℓ. (This is the same as EWM except the length is exactly ℓ, instead of r mod q.) The trivial algorithm for this problem is in time $O(n^\ell)$, but can we do better? To our knowledge, this problem has not been studied.

Another example is looking for edge-minimum walks that achieve constraints expressed by a finite semigroup. For instance, assume that each edge of the graph is labeled by a semigroup element, and that we want a walk whose evaluation in the semigroup achieves a specific target element of the semigroup, where evaluating the walk means multiplying the labels of its edges in the order that they are traversed. This problem generalizes the EWM problem (with lengths on edges), which uses the semigroup $\mathbb{Z}/q\mathbb{Z}$. An alternative way to phrase this problem is in the language of regular path queries (RPQs) mentioned in the introduction: fixing a regular language L on an alphabet Σ, and given a directed graph G with terminals s and t and with edges labeled by letters of Σ, find an edge-minimum subgraph G with an st-walk which evaluates to a word that belongs to L. For which fixed regular languages L can this problem be solved in polynomial time in G?

References

1 Noga Alon and Michael Krivelevich. Divisible subdivisions. *J. Graph Theory*, 98(4), 2021. doi:10.1002/jgt.22716.

2 Antoine Amarilli. Survey of results on the ModPath and ModCycle problems, 2024. doi:10.48550/arXiv.2409.00770.

3 Antoine Amarilli, Benoît Groz, and Nicole Wein. Edge-minimum walk of modular length in polynomial time, 2024. arXiv:2412.01614.

4 Esther M. Arkin, Christos H. Papadimitriou, and Mihalis Yannakakis. Modularity of cycles and paths in graphs. *J. ACM*, 38(2), 1991. doi:10.1145/103516.103517.

5 Guillaume Bagan, Angela Bonifati, and Benoît Groz. A trichotomy for regular simple path queries on graphs. *J. Comput. Syst. Sci.*, 108, 2020. doi:10.1016/J.JCSS.2019.08.006.

6 Andreas Björklund, Thore Husfeldt, and Petteri Kaski. The shortest even cycle problem is tractable. In *STOC*, 2022. doi:10.1145/3519935.3520030.

7 Archit Chauhan, Samir Datta, Chetan Gupta, and Vimal Raj Sharma. The even-path problem in directed single-crossing-minor-free graphs, 2024. doi:10.48550/arXiv.2407.00237.

8 Chandra Chekuri, Guy Even, Anupam Gupta, and Danny Segev. Set connectivity problems in undirected graphs and the directed Steiner network problem. *ACM Trans. Algorithms*, 7(2), 2011. doi:10.1145/1921659.1921664.

9 Rajesh Chitnis, Andreas Emil Feldmann, and Pasin Manurangsi. Parameterized approximation algorithms for bidirected Steiner network problems. *ACM Trans. Algorithms*, 17(2), 2021. doi:10.1145/3447584.

10 Fan R. K. Chung, Wayne Goddard, and Daniel J. Kleitman. Even cycles in directed graphs. *SIAM J. Discret. Math.*, 7(3), 1994. doi:10.1137/S0895480192225433.

11 Mariano P. Consens and Alberto O. Mendelzon. Graphlog: a visual formalism for real life recursion. In *PODS*, 1990. doi:10.1145/298514.298591.

12 Isabel F. Cruz, Alberto O. Mendelzon, and Peter T. Wood. A graphical query language supporting recursion. In *SIGMOD*, 1987. doi:10.1145/38713.38749.

13 Gianlorenzo D'Angelo and Esmaeil Delfaraz. Approximation algorithms for node-weighted directed Steiner problems. In *IWOCA*, 2024. doi:10.1007/978-3-031-63021-7_21.

14 Shagnik Das, Nemanja Draganić, and Raphael Steiner. Tight bounds for divisible subdivisions. *J. Combin. Theory Ser. B*, 165, 2024. doi:10.1016/j.jctb.2023.10.011.

15 Irit Dinur and Pasin Manurangsi. ETH-hardness of approximating 2-CSPs and directed Steiner network. In *ITCS*, 2018. doi:10.4230/LIPICS.ITCS.2018.36.

16 Ajit A Diwan. Cycles of weight divisible by k, 2024. doi:10.48550/arXiv.2407.01198.

17 Jon Feldman and Matthias Ruhl. The directed Steiner network problem is tractable for a constant number of terminals. *SIAM J. Comput.*, 36(2), 2006. doi:10.1137/S0097539704441241.

18 Andreas Emil Feldmann and Dániel Marx. The complexity landscape of fixed-parameter directed Steiner network problems. *ACM Trans. Comput. Theory*, 15(3–4), 2023. doi:10.1145/3580376.

19 Steven Fortune, John E. Hopcroft, and James Wyllie. The directed subgraph homeomorphism problem. *Theor. Comput. Sci.*, 10, 1980. doi:10.1016/0304-3975(80)90009-2.

20 Esther Galby, Sándor Kisfaludi-Bak, Dániel Marx, and Roohani Sharma. Subexponential parameterized directed Steiner network problems on planar graphs: A complete classification. In *ICALP*, 2024. doi:10.4230/LIPICS.ICALP.2024.67.

21 Jiong Guo, Rolf Niedermeier, and Ondrej Suchý. Parameterized complexity of arc-weighted directed Steiner problems. *SIAM J. Discret. Math.*, 25(2), 2011. doi:10.1137/100794560.

22 Xiao Hu and Stavros Sintos. Finding smallest witnesses for conjunctive queries. In *ICDT*, 2024. doi:10.4230/LIPICS.ICDT.2024.24.

23 Alpár Jüttner, Csaba Király, Lydia Mirabel Mendoza-Cadena, Gyula Pap, Ildikó Schlotter, and Yutaro Yamaguchi. Shortest odd paths in undirected graphs with conservative weight functions. *Discrete Appl. Math.*, 357, 2024. doi:10.1016/j.dam.2024.05.044.

24 Naonori Kakimura and Ken-ichi Kawarabayashi. Packing cycles through prescribed vertices under modularity constraints. *Adv. in Appl. Math.*, 49(2), 2012. doi:10.1016/j.aam.2012.03.002.

25 Ken-ichi Kawarabayashi, Stephan Kreutzer, O-joung Kwon, and Qiqin Xie. A half-integral Erdos-Posa theorem for directed odd cycles. In *SODA*, 2023. doi:10.1137/1.9781611977554.ch118.

26 Andrea S. LaPaugh and Christos H. Papadimitriou. The even-path problem for graphs and digraphs. *Networks*, 14(4), 1984. doi:10.1002/NET.3230140403.

27 Anna Lubiw. A note on odd/even cycles. *Discret. Appl. Math.*, 22(1), 1988. doi:10.1016/0166-218X(88)90125-4.

28 Wim Martens, Matthias Niewerth, and Tina Popp. A trichotomy for regular trail queries. *Log. Methods Comput. Sci.*, 19(4), 2023. doi:10.46298/LMCS-19(4:20)2023.

29 William McCuaig. Pólya's permanent problem. *Electron. J. Comb.*, 11(1), 2004. doi:10.37236/1832.

30 Zhengjie Miao, Sudeepa Roy, and Jun Yang. Explaining wrong queries using small examples. In *SIGMOD*, 2019. doi:10.1145/3299869.3319866.

31 Burkhard Monien. The complexity of determining a shortest cycle of even length. *Computing*, 31(4), 1983. doi:10.1007/BF02251238.

32 Zhivko Prodanov Nedev. Finding an even simple path in a directed planar graph. *SIAM J. Comput.*, 29(2), 1999. doi:10.1137/S0097539797330343.

33 Mehdy Roayaei and Mohammadreza Razzazi. Parameterized complexity of directed Steiner network with respect to shared vertices and arcs. *Int. J. Found. Comput. Sci.*, 29(7), 2018. doi:10.1142/S0129054118500302.

34 Neil Robertson, P. D. Seymour, and Robin Thomas. Permanents, Pfaffian orientations, and even directed circuits. *Ann. of Math. (2)*, 150(3), 1999. doi:10.2307/121059.

35 Ildikó Schlotter and András Sebő. Odd paths, cycles and t-joins: Connections and algorithms, 2022. arXiv:2211.12862.

36 Paul Seymour and Carsten Thomassen. Characterization of even directed graphs. *J. Combin. Theory Ser. B*, 42(1), 1987. doi:10.1016/0095-8956(87)90061-X.

37 Carsten Thomassen. Even cycles in directed graphs. *Eur. J. Comb.*, 6(1), 1985. doi:10.1016/S0195-6698(85)80025-1.

38 Carsten Thomassen. The even cycle problem for planar digraphs. *J. Algorithms*, 15(1), 1993. doi:10.1006/jagm.1993.1030.

39 Vijay V. Vazirani and Mihalis Yannakakis. Pfaffian orientations, 0-1 permanents, and even cycles in directed graphs. *Discret. Appl. Math.*, 25(1-2), 1989. doi:10.1016/0166-218X(89)90053-X.

40 Paul Wollan. Packing cycles with modularity constraints. *Combinatorica*, 31(1), 2011. doi:10.1007/s00493-011-2551-5.

41 Raphael Yuster and Uri Zwick. Finding even cycles even faster. *SIAM J. Discrete Math.*, 10(2), 1997. doi:10.1137/S0895480194274133.

Doubly Sub-Linear Interactive Proofs of Proximity

Noga Amir
Weizmann Institute of Science, Rehovot, Israel

Oded Goldreich
Weizmann Institute of Science, Rehovot, Israel

Guy N. Rothblum
Apple, Cupertino, CA, USA

Abstract

We initiate a study of *doubly-efficient* interactive proofs of *proximity*, while focusing on properties that can be tested within query-complexity that is significantly sub-linear, and seeking interactive proofs of proximity in which

1. The query-complexity of verification is significantly smaller than the query-complexity of testing.
2. The query-complexity of the honest prover strategy is not much larger than the query-complexity of testing.

We call such proof systems doubly-sublinear IPPs (dsIPPs).

We present a few doubly-sublinear IPPs. A salient feature of these IPPs is that the honest prover does not employ an optimal strategy (i.e. a strategy that maximizes the verifier's acceptance probability). In particular, the honest prover in our IPP for sets recognizable by constant-width read-once oblivious branching programs uses a distance-approximator for such sets.

2012 ACM Subject Classification Theory of computation → Interactive proof systems

Keywords and phrases Interactive Proof Systems, Interactive Proofs of Proximity, Query Complexity, Read Once Branching Programs, Sub-linear

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.6

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2024/143/

1 Introduction

This work combines the mind-frames of interactive proofs of *proximity* (i.e., IPPs) and *doubly-efficient* interactive proofs (de-IPs), while giving the notion of doubly-efficient a new meaning that focuses on the *query-complexity* of the honest prover strategy. Specifically, we focus on properties that can be tested by reading a small portion of the input (equiv., have query-complexity that is significantly sub-linear), and seek interactive proofs of proximity in which

1. The query-complexity of verification is significantly *smaller than* the query-complexity of testing.
2. The query-complexity of the honest prover strategy is *not much larger than* the query-complexity of testing.

In addition, we may seek analogous relations for the time-complexities; yet, we recall that, in the context of property testing, time-complexity is secondary to query-complexity (see [4, Sec. 1.3.1]).

A salient feature of (almost all) the IPPs that we present is that the honest prover does not employ an optimal strategy, because it cannot afford to read the entire input (per the second query-complexity condition). In particular, while an optimal prover strategy for the verifiers that we present achieves perfect completeness, our honest provers don't.


© Noga Amir, Oded Goldreich, and Guy N. Rothblum;
licensed under Creative Commons License CC-BY 4.0



1.1 Our focus: query-efficient generation of proofs of proximity

Focusing on properties that can be tested in sub-linear query-complexity, we initiate a study of interactive proofs of proximity in which verification requires less queries than testing whereas proving does not require much more queries than testing. We recall the wider context first.

Property testing. The focus of property testing[6, 22] is on approximate decision procedures that read small portion of their input (see [4] for an introduction to the subject). For a predetermined property (i.e., a set of valid objects), approximate decision means distinguishing between objects that have the property and objects that are far from any object having this property.[1] Such procedures, called testers, are probabilistic and obtain local views of the object by performing queries; that is, the object is seen as a function and the testers get oracle access to this function (and thus may be expected to read only part of the object).

The insistence on procedures that read a small portion of the input reflects envisioning applications in which the input is huge and fetching it entirely is infeasible. In some of these applications, one may afford computation time that is almost linear in the size of the input, although fetching the entire input is still deemed infeasible.

Proofs. A generic question is whether proofs can offer verification that is more efficient than the corresponding decision. In the context of property testing, this refers to interactive proofs of proximity (IPPs)[3, 21]. In such proof systems, *verification should require significantly less queries than testing*, and suitable notions of completeness and soundness should hold.[2] Specifically, *completeness* means that inputs that have the property should be accepted with high probability (when the prover uses an adequate strategy), whereas *soundness* means that inputs that are far from the property should be rejected with high probability (no matter what strategy is employed by the prover). Indeed, in the completeness condition we consider a honest prover, whereas in the soundness condition we consider a cheating one.

Doubly-efficient proofs. Seeking to utilize proof systems in reality prohibits the use of arbitrary honest provers. Such applications require that the honest prover strategy be relatively efficient, a requirement that is captured by the term *double-efficiency*, which refers to the complexities of both the verifier and the honest prover. Given our postulate that it is infeasible to fetch the entire input and our focus on query complexity, this means that the honest prover strategy should also have query complexity that is significantly smaller than linear. We call such proof systems doubly-sublinear IPPs (dsIPPs).

Note that dsIPPs may exist only for properties that can be tested using small query complexity. This is the case because the interaction between the verifier and the honest prover (along with the queries they make) can be emulated by the tester. Hence, we focus on properties that can be tested using small query complexity and ask *for which of these properties we can obtain dsIPPs*?

---

[1] Distances are measured according to the relative Hamming measure; that is, $x \in \{0,1\}^n$ is considered ε-far from $y \in \{0,1\}^n$ if x and y differ on more than $\varepsilon \cdot n$ locations.

[2] In addition, one requires low communication complexity. In particular, this does not allow the prover to send the entire input to the verifier (a possibility that is not relevant in the context of double-efficiency that we consider here).

We stress that the notion of double-efficiency employed here is different from the notion employed in the context of IPs (see[10, 19, 5]) and also in studies of IPPs that made reference to doubly-efficient IPPs (e.g.[20]). In these prior cases, the reference was to polynomial running time of the honest prover, whereas we focus on its query complexity and on the case that it is small (and in particular is sub-linear).

1.2 Our Main Results

We construct doubly-sublinear IPPs (dsIPPs) for several problems. Specifically, for any property that can be decided by a constant-width read-once oblivious branching program (ROOBP), for approximating the input's Hamming weight, for telling whether a given function is a permutation, and for a relaxation of the graph bipartiteness problem in the bounded-degree model.

As mentioned earlier, the honest prover strategies in (almost all) the dsIPPs that we present do not employ an optimal strategy, because they cannot afford to read the entire input. In particular, while optimal prover strategies for these IPPs achieves perfect completeness (i.e., the verifier always accepts inputs that have the property), our honest provers do not achieve perfect completeness.

1.2.1 Protocol for Read-Once Oblivious Branching Programs

We construct a dsIPP for any property that can be decided by constant-width read-once oblivious branching programs (ROOBPs). Recall that in such branching programs, in each layer, all vertices are labelled by the same input variable (which is the one being read), and each input variable labels the vertices of at most one layer. A branching program decides a property Π if it accepts all inputs in the property and rejects all inputs that are not in the property. See Section 4.1 for a fuller discussion on ROOBPs.

Newman [16] showed that any property that can be decided by a constant-width ROOBP has a tester that makes $\mathsf{poly}(1/\varepsilon)$ queries, where the exponent of the polynomial is linear in the width of the branching program. We construct a dsIPP for any such property, where the verifier's query complexity is only $O(1/\varepsilon)$, which is optimal.[3]

▶ **Theorem 1** (dsIPP for ROOBPs, informal). *Let Π be a property that can be decided by a constant-width ROOBP, and let n and ε denote the input length and the proximity parameter. Then, for every $r : \mathbb{N} \to \mathbb{N}$, there exists an $r(n)$-round dsIPP for Π such that the verifier has query complexity $O(1/\varepsilon)$, the honest prover has query complexity $\widetilde{O}(n^{1/r(n)}) \cdot \mathsf{poly}(r(n)/\varepsilon)$, and the communication complexity is $O(n^{1/r(n)} \cdot r(n) \cdot (\log n)/\varepsilon)$.*

See Corollary 20 for a full and formal statement. Evidently, our protocol allows for a trade-off between the number of rounds (on one hand) and the query complexity of the prover and the communication (on the other hand). In particular, a logarithmic number of rounds suffices for getting the prover's query complexity and the communication to depend only poly-logarithmically on n. The verifier's runtime can be linear in the communication complexity assuming access to a suitable procedure specifying the structure of the ROOBP (see below).[4] The prover's runtime depends on the computational complexity of tolerant testing for properties decided by ROOBPs, see below.

[3] Consider the set property $\{0^i 1^{n-i} : i \in [n]\}$, the input $x = 0^{n/2}1^{n/2}$ and a honest prover strategy that refers to the input x but is also invoked as a cheating strategy on a random input r that is at Hamming distance 2ε from x. Then, an $o(1/\varepsilon)$-query verifier cannot distringuish the two cases.

[4] We remark that for a general non-uniform ROOBP no verifier can have a sublinear running time: for example, one layer in the ROOBP might make it go into a state that rejects all inputs. Without reading the entire ROOBP, the verifier has no hope of detecting whether there is a "universal rejection" layer.

Our protocol builds on an IPP of Goldreich, Gur and Rothblum[7], in which the honest prover reads the entire input. On the other hand, the IPP of[7] works also for unbounded width (but its communication complexity grows logarithmically with the width). We note that our dsIPP is actually *tolerant* (see below), whereas this was not the case for the protocol of [7] (or the tester of [16], see below).

A reduction to tolerant testing. An interesting feature of our IPP is that the honest prover strategy is reduced to *tolerant* testing (or distance approximation) for a property that can be decided by an ROOBP of (at most) the same width and length. A *tolerant* tester [17] for a property Π gets proximity parameters $\varepsilon_c < \varepsilon_f$ and is supposed to accept (w.h.p.) if the input is ε_c-close to the property, and to reject (w.h.p.) if the input is ε_f-far from the property. The complexity is usually measured as a function of the gap $\varepsilon_f - \varepsilon_c$ (and possibly also of the input length).

Theorem 1 follows by plugging any tolerant tester for constant-width ROOBPs into our general reduction (which is stated in Theorem 2). While, as noted above, Newman's tester for ROOBPs[16] is not tolerant, we show that that it can be made tolerant. This contribution of the current work is presented in the full version of the article.

Our reduction of the prover's strategy to tolerant testing ROOBPs actually gives a *tolerant* dsIPP of ROOBPs, where *Tolerant IPPs* are defined analogously to tolerant testers.[5] Fixing the ROOBP under consideration, the reduction assumes that the honest prover has access to a distance approximator that gets an ROOBP of (at most) the specified width and length, and approximates the distance of an input from the corresponding property (i.e., the property decided by the ROOBP). We need the deviation of the distance approximation to be $O((\varepsilon_f - \varepsilon_c)/r)$, and its error probability to be sufficiently small. The approximator only needs to operate on "sub-ROOBPs" of the original ROOBP.

Before stating the reduction, we also specify the access the verifier and the prover need to the ROOBP in order to get sublinear *runtimes*. We assume that the prover and the verifier have (unit cost) access to two procedures that *specify the structure of the ROOBP* as follows. The first procedure, given two nodes u and v in the ROOBP, returns a bit indicating whether there is a path from u to v. The second procedure gets as input the index i of a layer and returns the vertices in that layer. (We emphasize that these procedures do not depend on the input to the ROOBP: they ignore edge labels and refer to the ROOBP as a directed graph.)

▶ **Theorem 2** (Reducing tolerant dsIPP to distance approximation for ROOBPs, informal)**.** *For any constant w, let Π_w be a property that can be decided by a width w ROOBP, and let $\varepsilon_c, \varepsilon_f$ denote the proximity parameters. Suppose that there exists a distance approximator for w-width ROOBP (and its subgraphs) with query-complexity $Q_w(\delta, \eta, n)$ and time-complexity $T_w(\delta, \eta, n)$, when δ denotes the approximation parameter, η denotes the error probability bound, and the ROOBP has length n. Then, for every $r : \mathbb{N} \to \mathbb{N}$, there exists an $r(n)$-round tolerant dsIPP for Π_w such that*

- *The verifier's query-complexity is $O(1/\delta)$, and the honest prover's query-complexity is $O(n^{1/r(n)} \cdot Q_w(\delta, \eta)/\delta)$, where $\delta = (\varepsilon_f - \varepsilon_c)/3r(n)$ and $\eta = \delta/O(n^{1/r(n)})$.*
- *The communication complexity is $O(n^{1/r(n)} \cdot r(n) \cdot (\log n)/\delta)$.*

[5] Specifically, if the input is ε_c-close to the property, then the honest prover should convince the verifier to accept (w.h.p.); but if the input is ε_f-far from the property, then the verifier should reject (w.h.p.) regardless of the prover's strategy. We stress that, while in the context of testing, tolerant testing and distance approximation are equivalent (under suitable parameters), this is not the case for dsIPPs (see Section 2 and Remark 8).

- *Assuming that the verifier and the prover have access to procedures specifying the structure of the ROOBP (see above), the verifier's runtime equals the communication complexity (up to constant factors), and the honest prover's runtime is* $O(n^{1/r(n)} \cdot r(n) \cdot \mathsf{T}_w(\delta, \eta, n)) \cdot \widetilde{O}(1/\delta)$.

See Theorem 19 for a full and formal statement. As discussed above, Theorem 1 is obtained from the above reduction by plugging in a distance approximator for constant-width ROOBPs, which we construct based on Newman's (non-tolerant) property tester. While the reduction of Theorem 2 only has polynomial dependence on the width of the ROOBP, Newman's tester and our tolerant version of it have an exponential dependence on the width of the ROOBP (which is the reason that Theorem 1 holds only for constant width).[6] See the details in Section 4.6.

1.2.2 Protocol for Hamming Weight

We construct a dsIPP for the Hamming weight (HW) problem, where the input is a string $x \in \{0,1\}^n$ and a claim about its Hamming weight. Note that Hamming weight cannot be computed by constant-width ROOBP [18], so the results of Section 1.2.1 do not give a dsIPP for HW.

Approximating the (relative) Hamming weight up to distance ε requires $O(1/\varepsilon^2)$ queries in the standalone setting. We construct a dsIPP with $O(1/\varepsilon)$ verifier queries (which is optimal).

▶ **Theorem 3** (dsIPP for Hamming weight, informal)**.** *Let n and ε denote the input length and the proximity parameter for the Hamming weight problem (HW). Then, for every $r : \mathbb{N} \to \mathbb{N}$, there exists an $r(n)$-round dsIPP for HW such that the verifier has query complexity $O(1/\varepsilon)$ and the honest prover's query complexity and its runtime are both $\widetilde{O}(n^{1/r(n)} \cdot r^3(n)/\varepsilon^3)$. Furthermore, the verifier's runtime and the communication complexity are both $O(n^{1/r(n)} \cdot r(n) \cdot \widetilde{O}(1/\varepsilon))$.*

See Theorem 10 for a full and formal statement. This protocol also obtains a trade-off between the number of rounds and the honest prover's runtime. Taking the number of rounds to be logarithmic in n gives a *honest prover with query complexity and runtime that are poly-logarithmic in n*. The honest prover's query complexity and runtime grow cubically with (the reciprocal of) the proximity parameter. We wonder whether this dependence can be improved to quadratic (which would be optimal [1]).

Rothblum, Vadhan and Wigderson [21] also constructed an IPP for Hamming weight, but did not have a sub-linear honest prover. Our HW protocol borrows ideas from the IPP for ROOBP of[7]. Indeed, one can get an IPP for HW directly from their ROOBP protocol, by applying it to the $O(n)$-width ROOBP that computes HW. However, their protocol does not have a sub-linear prover.

1.2.3 Protocol for PERM

For $n \in \mathbb{N}$, the set PERM consists of all permutations over $[n]$. Here we consider the problem of testing (resp., verifying) whether a function $f : [n] \to [n]$ is in PERM. Note that PERM has a tester of complexity $O(\sqrt{n/\varepsilon})$, which we outline below, whereas a query lower bound of $\Omega(\sqrt{n})$ follows from[13].[7] Furthermore, PERM has two different (1-round) IPPs, which were presented in[13] and[7], respectively. We review them next:

[6] In contrast, Theorem 2 holds also for varying width, but in that case the complexities have a multiplicative factor of poly(w).

[7] See also a direct proof in[23, Apdx A].

1. In the IPP of[13] the verifier selects uniformly at random a point $r \in [n]$, and sends r to the prover, who is supposed to return its f-preimage; the verifier accepts if and only if the prover's answer is mapped by f to r.
2. In the IPP of[7], the verifier selects uniformly at random a point $r \in [n]$, queries f on it, and sends $y \leftarrow f(r)$ to the prover, who is supposed to return its f-preimage; the verifier accepts if and only if the prover's answer equals r.
   (This IPP, unlike the previous one, utilizes prover-oblivious queries.)

In both cases, $f \in$ PERM is always accepted, whereas functions that are ε-far from PERM are rejected with probability $\Omega(\varepsilon)$. (In both cases, the protocol is repeated $O(1/\varepsilon)$ times to yield an IPP.)

More importantly, in both cases, the honest prover finds the required f-preimage by querying f on all points (or practically so). Our initial conjecture was that an IPP for PERM cannot have a verifier that uses $o(\sqrt{n})$ queries and an honest prover that makes $o(n)$ queries. Interestingly, this "working conjecture" is wrong.

▶ **Theorem 4** (dsIPP for PERM)**.** *For every $\alpha \in (0, 0.5)$, there exists a 1-round IPP for PERM that has a verifier that uses $O(n/\varepsilon)^{0.5-\alpha}$ queries and an honest prover that uses $O(n/\varepsilon)^{0.5+\alpha}$ queries.*

The communication and time complexity of both parties is $\widetilde{O}((n/\varepsilon)^{0.5+\alpha})$.

Sketch of the proof of Theorem 4. The straightforward tester for PERM consists of selecting $q = O(\sqrt{n/\varepsilon})$ random points in $[n]$, querying the function $f : [n] \to [n]$ on them, and rejecting if and only if collisions are found (among the f-images).

Evidently, any $f \in$ PERM is accepted with probability 1, whereas (as can be seen later) any f that is ε-far from PERM is rejected with high constant probability. A similar analysis for the soundness case implies that for any f that is ε-far from PERM, if we select $O(n/\varepsilon)^{0.5+\alpha})$ random points, then we expect to see $\Omega(n^{2\alpha})$ collisions and that these collisions involve $\Omega(n^{2\alpha})$ disjoint pairs of points. This leads us to the following protocol.

1. The verifier selects $p = O(n/\varepsilon)^{0.5+\alpha}$ (distinct) random points in $[n]$, denoted $s_1, ..., s_p$, and sends them to the prover,
2. The prover queries the input $f : [n] \to [n]$ on these p sample points, and sends the answers $(a_1, ..., a_p) \leftarrow (f(s_1), ..., f(s_p))$ to the verifier.
3. The verifier rejects if it sees a collision (i.e., if $a_i = a_j$ for some $i \neq j$).
4. Otherwise, it sub-samples $m = O(n/\varepsilon)^{0.5-\alpha}$ of the original points, queries f on each of these m samples, and accept if and only if all answers match the prover's answers (i.e., if $f(s_i) = a_i$ for the i's it sub-sampled).

Clearly, $f \in$ PERM is always accepted. Turning to the soundness analysis, we fix an arbitrary function f that is ε-far from PERM and let $C \stackrel{\text{def}}{=} \{x \in [n] : |f^{-1}(f(x))| > 1\}$ denote the set of points that form collisions under f. Then, $|C| > \varepsilon \cdot n$; actually, $|C| - |f(C)| > \varepsilon \cdot n$, because modifying f to a permutation requires changing its value on all but a single point in $f^{-1}(y)$ for every $y \in f(C)$.

An analogous consideration applies to the sample of p points, denoted $S = \{s_1, ..., s_p\}$, that is sent to the prover. Specifically, the answers provided by the prover must disagree with the values of f on at least $|S| - |f(S)|$ points, because for each $s \in S$ the prover must provide different answers to all the other points in S that are in $f^{-1}(f(s))$ (since otherwise the verifier rejects in Step 2). Towards upper-bounding $|f(S)|$, we observe that it is upper-bounded by $|f'(S)|$, where $f' : [n] \to [n]$ is defined such that for every $x, x' \in [n]$ it

holds that (i) $f'(x) \neq f'(x')$ whenever $f(x) \neq f(x')$ and (ii) for each x that forms a collision under f, the restriction of f' to $f^{-1}(f(x))$ is 2-to-1 except on at most one element (in case $|f^{-1}(f(x))|$ is odd). Now, we lower-bound $|S| - |f'(S)|$ by the number of collisions of S under f' (i.e., $|\{\{i,j\} \in \binom{[p]}{2} : f'(s_i) = f'(s_j)\}|$)

First note that the number of collisions of $[n]$ under f' is at least $\sum_{y \in f(C)} \lfloor |f^{-1}(y)|/2 \rfloor$, which is at least $|C|/3 > \varepsilon n/3$. Hence, with high probability over the choice of $S \in \binom{[n]}{p}$, we get $\Omega(\varepsilon \cdot p^2/n) = \Omega((n/\varepsilon)^{2\alpha})$ collisions of S under f'. In this case, there are $\Omega((n/\varepsilon)^{2\alpha})$ disjoint pairs of points in S such that the elements in each pair have the same f'-image (and hence also the same f-image).

Wishing to avoid rejection in Step 2, the prover must cheat on the values of the $\Omega((n/\varepsilon)^{2\alpha})$ foregoing points (in S), but (with high probability) at least one of these points will appear in the sub-sample that the verifier selects in Step 4 (since the sub-sample rate is $m/p = O(\varepsilon/n)^{2\alpha}$). Hence, in Step 4, when quering the function f on this sub-sample, the verifier detects this cheating and rejects (w.h.p.). This completes the proof of Theorem 4.

1.2.4 Protocol for Bipartiteness

We construct a dsIPP for a *relaxation* of graph bipartiteness in the bounded degree model. In this model (see, e.g.,[4, Sec. 9.1]), the input is an undirected n-vertex graph of constant maximum degree d. The input graph is represented by its incidence function $g : [n] \times [d] \to [n] \cup \{0\}$ such that $g(v,i)$ is the i-th neighbor of the vertex v (or 0 if v has less than than i neighbors). The prover and the verifier have query access to this function. The distance between two n-vertex graphs is the ratio (over $dn/2$) of the number of edges on whose presence or absence they disagree.

The promise problem we consider is that the input graphs are rapidly-mixing graphs; that is, graphs in which a random lazy walk of logarithmic length starting at any vertex reaches any other vertex with probability $\Theta(1/n)$. (An ℓ-step lazy walk is described by a ℓ-long sequence over $[2d]$ such that $i \in [2d]$ indicates a step that moves from the current vertex v to its i^{th} neighbor (in case v has at least i neighbors) and staying in place if v has less than i neighbors.)

▶ **Theorem 5** (dsIPP for bipartiteness). *Let n and ε denote the number of vertices and the proximity parameter. Then, there exists a 1-round dsIPP for bipartiteness on rapidly-mixing graphs in the bounded-degree graph model in which the verifier's query complexity is $O(\log(n)/\varepsilon)$, and the honest prover's query and time complexity are $\widetilde{O}(\sqrt{n}/\varepsilon)$. Furthermore, the verifier's runtime is polylog$(n)/\varepsilon$, and the communication complexity is $O(\log(n)/\varepsilon)$.*

Related work. Goldreich and Ron [9] showed a $\Omega(\sqrt{n})$ lower bound on the query complexity of testing bipartiteness.[8] A later work of the same authors[8] shows a tester with $\widetilde{O}(\sqrt{n}) \cdot \mathsf{poly}(1/\varepsilon)$ query complexity. We remark that their tester works for general graphs, without needing the rapidly-mixing condition. Rothblum, Vadhan and Wigderson [21] constructed an IPP for an intermediate relaxation of bipartiteness, where the YES case includes all the bipartite graphs, but the NO case includes only the graphs that are both far from bipartite and rapidly-mixing. In their IPP, the verifier's query complexity is $O(\log(n)/\varepsilon)$, but the

[8] Their lower bound, which is stated for general 3-regular graphs, also holds under the additional promise that the graphs are rapidly-mixing (see the exposition in [4, Sec. 9.3.1]). In particular, the distribution of NO cases is concentrated on expander graphs (see [4, Clm. 9.18.1]). A similar analysis shows that the YES cases are bipartite expanders (w.h.p.)

honest prover is not sub-linear: it has to read the entire graph. Our proof of Theorem 5 uses the [21] protocol, but shows a sublinear-time implementation for the honest prover, where the implementation works for graphs that are both *rapidly-mixing* and bipartite.

Proof of Theorem 5: The sublinear-time honest prover. We first recall the[21] protocol. In the basic test, the verifier picks a random vertex s, and performs a lazy random walk of length $\ell = O(\log(n))$, which is long enough to guarantee the "rapidly-mixing" condition (see above), reaching a vertex t. The verifier sends s and t to the prover, asks the prover to recover the parity of a (simple) path that leads from s to t, and accepts if and only if the answer equals the parity of the number of "real moves" in the lazy walk (i.e., moves in which the walk did not stay in the current vertex). If the graph is bipartite, then the prover, who can read the entire graph, can always answer correctly by checking whether s and t are on the same side of the bipartition. As shown in[21], if the graph is rapidly mixing and ε-far from bipartite, then a cheating prover will fail with probability $\Omega(\varepsilon)$. In order to reduce the soundness error (from $1 - \Omega(\varepsilon)$ to $1/3$), this basic test is repeated $r = O(1/\varepsilon)$ times in parallel (and the verifier accepts if and only if it accepted in all invocations of the basic test).

We construct a sub-linear time honest prover for the basic test. Upon receiving the vertices s and t, the prover performs $w = O(\sqrt{n} \cdot \log(1/\varepsilon))$ independent random walks of length ℓ starting at s, as well as w independent random walks of length ℓ starting at t. Let W_s (resp., W_t) denote the set of vertices traversed in (the union of all) the walks that started at s (resp., t). The prover checks if there is a vertex v in the intersection of W_s and W_t (i.e., a vertex encounted in a walk that started at s as well as in a walk that started at t). If so, then the prover has found a walk from s to t (going through v), and it replies with the parity of the number of real moves on this walk.

Soundness follows directly from the soundness of the [21] protocol (since we did not change the verifier). We now argue for completeness (alas not perfect completeness). If the graph is bipartite, then all paths from s to t have the same parity. Thus, whenever the prover finds a path from s to t (i.e., when the intersection is non-empty), then it returns the correct parity. It remains to show that in each of execution of the basic test (i.e., for each s and t sent by the verifier), the intersection of W_s and W_t is non-empty with probability at least $1 - (1/3r)$. We show this in Claim 6, which is where we use the condition that the graph is rapidly-mixing, and it follows that the verifier accepts in all r repetitions with probability at least $2/3$.

▷ Claim 6. If the graph is rapidly-mixing, then, for every two vertices $s, t \in [n]$, the probability, over the honest prover's random walks, that $W_s \cap W_t = \emptyset$ is at most $1/3r$.

Proof. We focus on the sets $W'_s \subset W_s$ and $W'_t \subset W_t$ of *terminal* vertices reached by the random walks (i.e., the final vertex in each walk), and show that these subsets intersect with probability at least $1 - (1/3r)$. Using the rapidly-mixing condition, observe first that taking $w = O(\sqrt{n} \cdot \log(1/\varepsilon)) = O(\sqrt{n} \log r)$ independent walks from s, with probability at least $1 - (1/6r)$, the number of distinct vertices reached (i.e., the size of the set W'_s) is $\Omega(\sqrt{n})$. Assuming that this event occurs, and recalling that each of the random walks starting at t is rapidly-mixing, we infer that, with probability at least $\Omega(1/\sqrt{n})$, such a walk terminates in a vertex that resides in W'_s. Recalling that there are $O(\sqrt{n} \cdot \log r)$ walks starting at t, the probability that they all land outside of W'_s is thus at most $1/6r$. The claim follows. ◁

1.3 Further Related Work

Goldwasser, Rothblum, Shafer and Yehudayoff [11] studied interactive proofs for approximate verification of the results of a machine-learning computation on an unknown underlying distribution. The verifier and the honest prover both have sampling access to the unknown distribution (in some of their results the honest prover also has membership queries, see also [12]). While their model is quite different from ours, there is a similarity in spirit: if we think of the unknown distribution as a "huge input" to the proof system, then they are also concerned with proof systems where verification is more efficient than the computation being verified (a learning problem), and proving is not much more expensive than performing the same computation. In particular, proving is sub-linear in the size of the "input" (the unknown distribution). The settings, however, are quite different: both in the access to the input (querying an unknown function vs. sampling from an unknown distribution), and in the types of tasks studied (approximate decision vs. machine learning). We stress that other recent works on verifying properties of unknown distributions [2, 14, 15] do *not* study *honest provers with sublinear sample complexity.*

1.4 Technical Overview

In this section provide overviews of our dsIPPs for Hamming Weight and for properties that are decidable by ROOBPs. Recall that the dsIPPs for PERM and for a relaxed version of Bipartiutness were sketched in prior sections.

We start by outlining our IPPs for Hamming Weight (HW), and then extend its underlying ideas to obtain IPPs for ROOBPs. Both IPPs follow an abstract idea that underlies the IPPs of[21, 7], but deviate from it in a significant manner. Loosely speaking, the IPPs of[21, 7] rely on partitioning the original property to sub-properties that refer to parts of the original input. However, while the IPPs of[21, 7] call for an honest prover strategy that finds optimal partitions of the property to sub-properties, we shall use partitions that are sub-optimal, because these sub-optimal partitions can be found more efficiently (by the honest prover).

IPPs for Hamming Weight (HW). Our r-round IPPs proceed by recursion, where for a parameter k (to be set to $n^{1/r}$), the current input x is partitioned to k equal-length blocks, denoted $x_1, ..., x_k$. Clearly, the Hamming weight of x, denoted $\mathtt{wt}(x)$, equals the sum of the Hamming weights of the x_i's (i.e., $\sum_{i\in[k]} \mathtt{wt}(x_i)$). Hence, wishing to prove that $\mathtt{wt}(x)$ equals w, an optimal prover determines the corresponding weights of the x_i's, denoted $w_1, ..., w_k$, sends the w_i's to the verifier, who selects a random i, and the parties proceed to prove that the weight of x_i equals w_i. (Indeed, after r iterations, the verifier can check the claim, which refers to a single bit, by making a single query.) Needless to say, determining these w_i's requires reading the entire input x, which we want to avoid. Instead, we consider a query-efficient honest prover that obtains approximations of the desired w_i's (or obtains the exact value when $|x| = k$, which happens after $r-1$ recursion steps). This requires relaxing the verification procedure so that it does not reject in case the sum $\sum_i w_i$ does not equal w (but is rather only close to it).

As in[21], the soundness analysis relies on the fact that if $|\mathtt{wt}(x) - w| > \Delta$, then, for every sequence of w_i's such that $\sum_{i\in[k]} w_i = w$, it holds that $\sum_{i\in[k]} |\mathtt{wt}(x_i) - w_i| > \Delta$; equivalently, if x is ε-far from having weight w, then, for every sequence of w_i's such that $\sum_{i\in[k]} w_i = w$, on the average, x_i is ε-far from having weight w_i. Note, however, that if we allow $\sum_{i\in[k]} w_i$ to deviate from w by say $\varepsilon' \cdot |x|$, then, on the average, x_i is $(\varepsilon - \varepsilon')$-far from having weight w_i. Hence, at the bottom of the recursion, the expected distance of single bits from their claimed weight is $\varepsilon - r \cdot \varepsilon'$, which means that the verifier rejects with such probability. Using $\varepsilon' = \varepsilon/2r$ and repeating the protocol $O(1/\varepsilon)$ times, we obtain the desired IPP.

IPPs for constant-width ROOBP. As in the case of HW, we wish to proceed by recursion. Indeed, in the r-round IPP of[7], on current input x, an optimal prover determines the path in the current ROOBP that accepts x, which (as before) is partitioned to equal-length strings $x_1, ..., x_k$. This path defines k sub-ROOBPs that each accept the corresponding x_i's. Alas, determining these sub-ROOBPs requires reading the entire input x, which we want to avoid.

Using the fact that the ROOBP has bounded width, it follows that, for each $i \in [k]$, there is a bounded number of sub-ROOBPs (i.e., the square of the width bound) such that x_i is accepted by (at least) one of them. Using a distance approximator for (constant-width) ROOBPs, the honest prover can find, for each $i \in [k]$, a sub-ROOBP such that x_i is close to being accepted by this sub-ROOBP. This means that the foregoing description has to be modified: In subsequent iteration of the recursion it does not necessarily hold that the current x is accepted by the current ROOBP; it is only the case that the current x is close to being accepted by the current ROOBP. But in this case, for $x = (x_1, ..., x_k)$, it does not necessarily hold that each x_i is close to being accepted the corresponding sub-ROOBPs, but it is rather the case that their average distance from these sub-ROOBPs is small; that is, if x is ε-close to the current ROOBP, then for some $\varepsilon_1, ..., \varepsilon_i$ such that $\sum_{i\in[k]} \varepsilon_i/k = \varepsilon$, each x_i is ε_i-close to a corresponding sub-ROOBP, but it is not necessarily the case that $\varepsilon_i \approx \varepsilon$ for each $i \in [k]$.

Recall that the honest prover does not find these ε_i's, but rather finds their approximate values (as well as the corresponding sub-ROOBPs). Specifically, for width bound b, the honest prover considers an auxilary graph G with $k+1$ layers, index $0, 1, ..., k$, such that the i^{th} layer of G contains the vertices that are at the $i \cdot k$ layer of the current ROOBP. Each pair of vertices in adjacent layers of G represents a possible sub-ROOBP, where pairs in layers $i-1$ and i of G represent a sub-ROOBP that reads x_i. For each such pair, the honest prover estimates the distance of x_i from being accepted by the corresponding sub-ROOBP, where these estimates are obtained by invoking the distance-approximator (for ROOBPs). Using a shortest path algorithm, the honest prover finds sequence of intermediate vertices $(v_1, ..., v_{k-1})$ in G such that the average distance of the x_i's from the corresponding sub-ROOBPs, denoted $(B_1, ..., B_k)$, is minimal, and sends $(v_1, ..., v_{k-1})$ to the verifier along with the corresponding distances. (As in the case of HW, the verifier will select $i \in [k]$ uniformly at random, and the parties will proceed to prove the corresponding claim (i.e., that x_i is ε_i-close to being accepted by the corresponding sub-ROOBP).)

2 Preliminaries and Definitions

For strings $x, y \in \{0,1\}^n$, the relative Hamming distance between x and y is the fraction of coordinates in which they disagree (we often refer to this as the *distance* for short). If this distance is at most ε, then we say that x is ε-close to y, and otherwise we say that x is ε-far from y. We define the distance of x from a (non-empty) set $S \subseteq \{0,1\}^n$ as its distance from the closest y in S, and we define the string being ε-close and ε-far from the set analogously. We extend these definitions from strings to functions by identifying a function with its truth table.

A property Π (or a language) is a set of strings of varying lengths (i.e. in $\{0,1\}^*$). The approximate decision problem for Π is deciding, for specified proximity parameters, whether an input string is close or far from the property. See [4] for an introduction to property testing: the field that studies algorithms of sublinear query complexity for such approximate decision problems.

▶ **Definition 7** (Interactive Proof of Proximity (IPP, tolerant)). *An IPP is a protocol between two probabilistic parties, a prover P and a verifier V who both get an input length n and a joint input $x \in \{0,1\}^n$. The verifier has query access to x (and explicit access to the input length n). The parties interact and at the end of the interaction the verifier accepts or rejects. The protocol is a (tolerant) IPP for a property Π and for proximity parameters $\varepsilon_c, \varepsilon_f : \mathbb{N} \to \mathbb{R}$ if for every input length n and every input $x \in \{0,1\}^n$:*

- ***Completeness:*** *If x is $\varepsilon_c(n)$-close to the property Π (in relative Hamming distance), then the verifier, after interacting with the prover P, accepts with probability at least $2/3$ (the probability is over the prover's and the verifier's coin tosses). If the verifier, interacting with the honest prover, accepts every input in the language with probability 1 then we say that the IPP has* perfect completeness.
  In this work we focus on IPPs where the honest prover only has query access to the input x (similarly to the verifier, it gets the input length n as an explicit input).
- ***Soundness:*** *If x is $\varepsilon_f(n)$-far from the property Π, then for every cheating strategy P^*, the verifier V, after interacting with the prover P^*, rejects with probability at least $2/3$ (the probability is over the verifier's coin tosses, w.l.o.g. the cheating prover can be taken to be deterministic, and can have explicit access to the entire input).*

A non-tolerant IPP is one where $\varepsilon_c = 0$. In this case, we refer to a single proximity parameter $\varepsilon = \varepsilon_f$ (the parameter ε_c is implicit). Testers are viewed as a special case of IPPs in which there is no prover (or no interaction with it).

The protocol's complexity measures include the (honest) prover's and the verifier's query complexities and runtimes, the communication complexity and the round complexity (the number of back-and-forth communication rounds). These complexities are typically measured as a function of the gap $(\varepsilon_f(n) - \varepsilon_c(n))$ between the proximity parameters and of the input length.

Our focus in this work is on protocols where the query complexities of the verifier and the honest prover are as small as possible. In particular, we focus on properties that have testers of sublinear query complexity and require that the verifier's query complexity should be smaller than the tester's. The prover's query complexity should be as close as possible to the tester's. We refer to these as *doubly sub-linear* IPPs (dsIPPs, see the discussion in the Introduction).

▶ Remark 8. We emphasize that, while in the context of testing, tolerant testing and distance approximation are equivalent [17] (under suitable parameters), this is not the case for IPPs. The reason for this divergence is that a tolerant IPP only provides a one-sided guarantee: it can affirmatively convince the verifier of *upper bounds* on the input's distance from the property, but it does not convince the verifier of a *lower bound* on the distance.

▶ Remark 9. We remark that in the study of standard IPPs (where the honest prover can read the entire input), if we allow linear communication complexity, then the query complexity can always be tiny. This is because the prover can send the entire input to the verifier. The verifier receives this alleged input and verifies that it is close to the real input by checking consistency for a few random locations (these are the only queries made to the real input). If the alleged input is close to the real input, then the verifier can simply check if the alleged input is in the property (or, for tolerant protocols, that it is at the appropriate distance from the property). For dsIPPs it is not clear that linear communication always allows for a sublinear verifier. In particular, the aforementioned strategy cannot be utilized because the honest prover doesn't know the entire input and cannot send it. Thus, dsIPPs with linear (or even polynomial) communication may be an interesting object for further study.

3 dsIPP for the Hamming Weight Problem

In this section, we present a doubly sub-linear IPP (dsIPP) for the Hamming Weight Problem. Specifically, given a claimed weight $\sigma \in \mathbb{N}$, a proximity parameter $\varepsilon > 0$ and an oracle access to an input $x \in \{0,1\}^n$, we present an r-round IPP, with $r = \log(n)$ and with poly-logarithmic communication complexity, in which V verifies that $\mathtt{wt}(x) = \sigma$ using query complexity of $O(\frac{1}{\varepsilon})$, the honest prover P uses poly-logarithmic query complexity. We note that we use r to represent the number of (pairs) of message exchanges (and not the total number of messages sent).

First, in Section 3.1, we present a standard IPP for the Hamming Weight Problem. This IPP (which is not doubly sub-linear) uses a divide-and-conquer approach, and in it the prover P computes exact weights for sub-sequences of the input x. Then, in Section 3.2, we present a warm-up towards a dsIPP: We amend the standard IPP by allowing P to approximate the Hamming weight of the sub-sequences, and relaxing the accepting conditions of V in a way that doesn't compromise the completeness and soundness of the protocol. Unfortunately, in this IPP the prover P still has query complexity of $\Omega(n)$. Lastly, in Section 3.3, we present the actual dsIPP, which applies the IPP presented in Section 3.2 recursively, thus improving the query complexity of P without compromising the query complexity of V.

3.1 The Standard IPP

Given some σ, we want to verify that $\mathtt{wt}(x) = \sigma$. First, we observe that when we partition x to k consecutive equally-length parts, $x_1, ..., x_k$, the following holds:

- On the one hand, if $\mathtt{wt}(x) = \sigma$, then $\sum_i \mathtt{wt}(x_i) = \sigma$.
- On the other hand, if $|\mathtt{wt}(x) - \sigma| > \varepsilon n$, then for any $(\sigma_1, ..., \sigma_k)$ such that $\sum_i \sigma_i = \sigma$, it holds that $\sum_i |\mathtt{wt}(x_i) - \sigma_i| > \varepsilon n$.

Given the above, we consider the following IPP: P sends $(\sigma_1, ..., \sigma_k) \leftarrow (\mathtt{wt}(x_1), ..., \mathtt{wt}(x_k))$ to V. Then, V verifies that $\sum_i \sigma_i = \sigma$, and if so, selects at random $i \in [k]$, and verifies that $\mathtt{wt}(x_i) = \sigma_i$ (by reading the entire x_i). Note that if $\mathtt{wt}(x) = \sigma$ and V interacts with P, then for every i, it holds that $\mathtt{wt}(x_i) = \sigma_i$, and the verifier always accepts. However, if $|\mathtt{wt}(x) - \sigma| > \varepsilon n$ and $\sum_i \sigma_i = \sigma$, it holds that $\sum_i \frac{\mathtt{wt}(x_i) - \sigma_i}{k} > \varepsilon \cdot \frac{n}{k}$. Then, the average deviation of the claimed weight of x from the actual weight of x translates to a deviation on a corresponding fraction of random i's, hence guarantees that the verifier rejects with probability at least ε. To increase the rejection probability, we can repeat this procedure for $O(\frac{1}{\varepsilon})$ times.

The query complexity of the verifier is the length of a single x_i times the number of repetitions, which gives $O(\frac{n}{\varepsilon \cdot k})$. However, to compute $(\mathtt{wt}(x_1), ..., \mathtt{wt}(x_k))$, the prover must access to the entire input x, which yields query complexity of n.

3.2 A Warm-Up towards a dsIPP

In this section, we aim to achieve poly-logarithmic query complexity for the honest prover P without compromising the query complexity of the verifier V: We amend the interaction described in Section 3.1 by allowing P to *approximate* the Hamming weight of the sub-parts of x. However, at the end of this section, we explain why P still has query complexity of $\Omega(n)$. In Section 3.3, we show how to solve this issue and present the actual dsIPP.

First, we observe that for a deviation parameter $\delta > 0$ and an error probability parameter $\eta > 0$, there exists a (δ, η)-approximator for the Hamming weight of $x \in \{0,1\}^n$ with query complexity $O(\frac{\log(\frac{1}{\eta})}{\delta^2})$ and error probability η: Given the input x and the deviation parameter

δ, the approximator chooses, uniformly at random, $O(\frac{\log(\frac{1}{\eta})}{\delta^2})$ indices of x, and outputs the (normalized) number of indices i for which $x[i] = 1$. Using Chernoff bound, we can show that with probability at least $1 - \eta$, the output of the approximator deviates from $\mathtt{wt}(x)$ by at most δn.

Based on this approximator, we amend the interaction such that the prover provides *approximated* weights to the verifier, by using the approximator with some predefined error probability $\eta > 0$ and deviation parameter $\delta > 0$ (we fix both parameters later). That is, P sends $(\sigma_1, ..., \sigma_k)$ to V, such that for every $i \in [k]$, σ_i is a (δ, η)-approximation for $\mathtt{wt}(x_i)$. We also change the verifier's accepting conditions in the interaction, so that V allows this deviation (and doesn't reject P's approximations). That is, V now verifies that $|\sigma - \sum_i \sigma_i| \leq \delta n$, chooses $i \in [k]$ uniformly at random, and verifies that $|\mathtt{wt}(x_i) - \sigma_i| \leq \frac{\delta \cdot n}{k}$ (by reading the entire x_i). Similar to the standard IPP, we repeat the protocol for $O(\frac{1}{\varepsilon})$ times.

3.2.1 Correctness

The completeness of the protocol still holds (but with a completeness error): If $\mathtt{wt}(x) = \sigma$ and we interact with P, then for a single invocation of the protocol the following holds: For every $i \in [k]$, with probability at least $1 - k \cdot \eta$, it holds that $|\sigma_i - \mathtt{wt}(x_i)| \leq \frac{\delta \cdot n}{k}$, and therefore also $\sum_i |\mathtt{wt}(x_i) - \sigma_i| \leq \delta n$, and the verifier accepts. Since we invoke the protocol for $O(\frac{1}{\varepsilon})$ times, then the verifier accepts in all invocations with probability at least $1 - t \cdot \eta$ where $t = O(\frac{k}{\varepsilon})$. Thus, we can set $\eta \leq 0.01 \cdot \frac{1}{t} = O(\frac{\varepsilon}{k})$, and the verifier accepts with high constant probability in all invocations.

To show the soundness of the protocol also holds, we observe that if $|\mathtt{wt}(x) - \sigma| > \varepsilon n$, then for any $(\sigma_1, ..., \sigma_k)$ such that $|\sigma - \sum_i \sigma_i| \leq \delta n$, it holds that $\sum_i |\mathtt{wt}(x_i) - \sigma_i| > (\varepsilon - \delta)n$. Then, the average deviation of the claimed weight of x from the actual weight of x, beyond the allowed deviation, translates to a deviation on a corresponding fraction of random i's. Since the verifier rejects if $|\mathtt{wt}(x_i) - \sigma_i| > \frac{\delta n}{k}$, we can set $\delta = \frac{\varepsilon}{3} < \frac{\varepsilon}{2}$ and infer that the verifier still rejects with probability at least $\Omega(\varepsilon)$ in a single invocation. Since we invoke the protocol for $O(\frac{1}{\varepsilon})$ times, we get that the verifier rejects with high constant probability in at least one of the invocations.

3.2.2 Query Complexity

Now, let us analyze the query complexity of V and P:

- The query complexity of V is $O(\frac{n}{\varepsilon k})$, because we read a sub-sequence of the input of length $\frac{n}{k}$ for $O(\frac{1}{\varepsilon})$ repetitions.
- The query complexity of P is $O(\frac{k \cdot \log(\frac{k}{\varepsilon})}{\varepsilon \cdot \delta^2})$, since we invoke the approximator with error probability parameter $\eta = O(\frac{\varepsilon}{k})$ for k times in each of the $O(\frac{1}{\varepsilon})$ repetitions.

Since we can $(\delta, 0.1)$-approximate the Hamming weight of x without any interaction using query complexity $O(\frac{1}{\delta^2})$, we can perform ε-test for the Hamming weight of x using query complexity $O(\frac{1}{\varepsilon^2})$. Therefore, for the IPP to be meaningful, we want the query complexity of V to be $o(\frac{1}{\varepsilon^2})$. To achieve this in our setting, we must require $k = \omega(\varepsilon \cdot n)$. However, this causes the query complexity of P to be $\Omega(n)$, whereas we want our IPP to be doubly sub-linear. In the next section, we extend the foregoing IPP by applying it recursively, and show that we can get both query complexity of $o(\frac{1}{\varepsilon^2})$ for V, **and** poly-logarithmic query complexity for P.

3.3 The Actual dsIPP

In Section 3.2, we aim to achieve poly-logarithmic query complexity for the honest prover P, without compromising the query complexity of the verifier V. Unfortunately, for reasons explained at Section 3.2.2, P still has query complexity of $\Omega(n)$.

To solve this, we extend the IPP presented in Section 3.2 by applying it recursively: That is, after V selects at random $i \in [k]$, instead of directly computing the Hamming weight of x_i, both parties recursively run the protocol on x_i with the claimed Hamming weight σ_i, and do it for r rounds. Only in the base of the recursion, V computes the Hamming weight of the input (at this level), and accepts accordingly. This gives us an improvement in the query complexity of the verifier, as the length of the input after r rounds is $(\frac{n}{k^r})$, while the number of times the prover needs to run the approximator grows only to $k \cdot r$ (in every repetition). To make sure that the query complexity of V is $o(\frac{1}{\varepsilon^2})$, we set $k = n^{\frac{1}{r}}$.

Indeed, we now allow the prover to deviate from the claimed weight for r times: At each of the recursion levels, as well as in the base level. Thus, to make sure that soundness still holds (i.e. the total allowed deviation is not too big), we set the verifier's allowed deviation (i.e. δ) to be smaller (by a factor of r).

To make sure that completeness still holds, we first change the error probability parameter of the approximator. P now runs the approximator for a total of $t = O(\frac{k \cdot r}{\varepsilon})$ times, and so we set $\eta = 0.01 \cdot \frac{1}{t} = O(\frac{\varepsilon}{k \cdot r})$. In addition, we observe the following: Assume V interacts with P, with input x and weight parameter σ, such that $\mathtt{wt}(x) = \sigma$. Then, when we are not in the first recursion level, then the weight parameter provided to the protocol is the approximated weight provided by P in the previous recursion level. Therefore, the weight parameter in the current recursion level might deviate from the actual weight of the input in this level. Since P also performs weight approximations in the current recursion level, we need to account for both deviations. Details follow.

▶ **Protocol 1.** *dsIPP for the Hamming Weight Problem*

- ***Query Access Input:*** $x \in \{0,1\}^n$
- ***Deviation Parameter:*** $\varepsilon > 0$
- ***Other Parameters:*** *Weight* $\sigma \in \mathbb{N}$*, initial number of rounds* r_0 *and remaining number of rounds* r

1. *Set* $k = n^{\frac{1}{r_0}}$*,* $\delta = \frac{\varepsilon}{3r_0}$*, and* $\eta = O(\frac{\varepsilon}{k \cdot r_0})$*.*
2. *V: If* $r = 0$ *then accept iff* $|\mathtt{wt}(x) - \sigma| \leq \frac{\delta n}{2}$*.*
3. *P:*
   a. *For every* $i \in [k]$*, approximate* $\mathtt{wt}(x_i)$ *up to a deviation factor of* $\frac{\delta}{2}$ *with probability at least* $1 - \eta$*: Choose uniformly at random* $m = O(\frac{log(\frac{1}{\eta})}{\delta^2})$ *indices of* x_i *and set the approximation* σ_i *as the estimated fraction of indices* j *for which* $x_i[j] = 1$*.*
   b. *Send* $(\sigma_1, ..., \sigma_k)$ *to* V*.*
4. *V:*
   a. *Verify that* $|\sigma - \sum_i \sigma_i| \leq \delta n$*, otherwise reject.*
   b. *Choose uniformly at random* $i \in [k]$ *and send* i *to* P*.*
5. *Both parties recursively invoke the protocol with input* x_i*, deviation parameter* ε*, weight* σ_i*, and remaining number of rounds* $r - 1$*.*

Repeat the protocol for $O(\frac{1}{\varepsilon})$ *times, and accept iff* V *accepts in all repetitions.*

We show that Protocol 1 is a dsIPP for the Hamming Weight Problem: In Section 3.3.1, we show the correctness of the protocol, and in Section 3.3.2, we present the query complexity of the protocol.

3.3.1 Correctness

Completeness. Assume $\mathtt{wt}(x) = \sigma$ and V interacts with P. We observe that at any recursion level (but the base level), the following claim holds: If our input is x' and our weight parameter is σ' such that $|\mathtt{wt}(x') - \sigma'| \leq \frac{\delta|x'|}{2}$, then with probability at least $1 - \eta \cdot k$, the prover P sends $(\sigma'_1, ..., \sigma'_k)$ to V such that the following holds:

1. The verifier doesn't reject in Step $4(a)$ (i.e., $|\sigma' - \sum_i \sigma'_i| \leq \delta|x'|$).
2. The protocol is recursively invoked with some x'_i and σ'_i for which $|\mathtt{wt}(x'_i) - \sigma'_i| \leq \frac{\delta|x'|}{2}$.

The claims follows from the fact that P runs the approximator on every x'_i with deviation parameter $\frac{\delta}{2}$ and error probability parameter η. Then, since we start with $\mathtt{wt}(x) = \sigma$, with probability at least $1 - \eta \cdot k \cdot r_0$, the verifier doesn't reject in Step $4(a)$ in all the r_0 recursion levels. By the second item of the claim, the base of the recursion is also invoked with an input x' and weight parameter is σ' such that $|\mathtt{wt}(x') - \sigma'| \leq \frac{\delta|x'|}{2}$, and the verifier accepts. Since we repeat the protocol for $O(\frac{1}{\varepsilon})$ times, we get that the verifier accepts in all invocations with probability at least $1 - t \cdot \eta$ where $t = O(\frac{k \cdot r_0}{\varepsilon})$. Then, since we set $\eta = 0.01 \cdot \frac{1}{t} = O(\frac{\varepsilon}{k \cdot r_0})$, we get that V accepts with high constant probability in all invocations.

Soundness. Assume $|\mathtt{wt}(x) - \sigma| > \varepsilon n$. We observe that in each recursion level, we lose at most an additive factor of δ. Since we set $\delta = \frac{\varepsilon}{3r_0}$, we get that after r_0 rounds, we are still at distance at least $\frac{\varepsilon}{2}$ from a valid assertion. Thus, the verifier rejects with probability at least $\Omega(\varepsilon)$ *in each invocation*, and therefore rejects with high constant probability in at least one of the $O(\frac{1}{\varepsilon})$ invocations.

3.3.2 Computational Complexity

We present the query, communication and time complexity of Protocol 1. Let us start by analyzing the query complexity of V (denoted Q_V) and P (denoted Q_P):

- $\mathsf{Q}_V = O(\frac{n}{\varepsilon k^{r_0}}) = O(\frac{1}{\varepsilon})$, because we read a sub-sequence of the input of length $\frac{n}{k^{r_0}} = 1$ for $O(\frac{1}{\varepsilon})$ times.
- Q_P is the query complexity of the approximator times the number of calls P makes to the approximator in all repetitions:
  - The query complexity of the approximator is $O(\frac{r_0^2 \cdot \log(\frac{k \cdot r_0}{\varepsilon}))}{\varepsilon^2})$, because P invokes the approximator with error probability parameter $\eta = O(\frac{\varepsilon}{k \cdot r_0})$ and deviation parameter $\delta = O(\frac{\varepsilon}{r_0})$.
  - P calls the approximator for $k \cdot r_0$ times in each of the $O(\frac{1}{\varepsilon})$ repetitions.

  Since we set $k = n^{\frac{1}{r_0}}$, we get that $\mathsf{Q}_P = \tilde{O}(\frac{n^{\frac{1}{r_0}} \cdot r_0^3}{\varepsilon^3})$.

Hence, regardless of how we set r_0, the query complexity of the verifier is $O(\frac{1}{\varepsilon}) = o(\frac{1}{\varepsilon^2})$. Indeed, the larger we set r_0, the smaller Q_P gets, but this causes the round complexity to grow. Specifically, to minimize Q_P, we can set $r_0 = \log(n)$, and we get $\mathsf{Q}_P = \tilde{O}(\frac{\log^3(n)}{\varepsilon^3})$.

Now, the communication of Protocol 1 is $O(\frac{k \cdot r_0}{\varepsilon} \cdot \log(\frac{r_0}{\varepsilon}))$: In each of the $O(\frac{1}{\varepsilon})$ repetitions there are r_0 rounds, and in each round the prover P sends k weight approximations that can be represented by $O(\log(\frac{r_0}{\varepsilon}))$ bits (because the approximations are up to a factor of $\delta = O(\frac{\varepsilon}{r_0})$). Again, since we set $k = n^{\frac{1}{r_0}}$, we can set $r_0 = \log(n)$ and get poly-logarithmic communication complexity.

Lastly, the time complexity of the honest prover (denoted T_P) is the same as its query complexity (i.e., $\mathsf{T}_P = \mathsf{Q}_P$), whereas the time complexity of the verifier (denoted T_V) is $O(\frac{k \cdot r_0}{\varepsilon} \cdot \log(\frac{r_0}{\varepsilon}))$, because in each of the $O(\frac{1}{\varepsilon})$ repetitions there are r_0 rounds, and in each round the verifier performs simple calculations on $O(k)$ values that can be represented by $O(\log(\frac{r_0}{\varepsilon}))$ bits.

We conclude with the following theorem, that summarizes what we achieved in this section.

▶ **Theorem 10.** *For every function $r : \mathbb{N} \to \mathbb{N}$, there exists an $r(n)$-round dsIPP for the Hamming Weight Problem with $Q_V = O(\frac{1}{\varepsilon})$, $Q_P = \tilde{O}(\frac{n^{\frac{1}{r(n)}} \cdot r(n)^3}{\varepsilon^3})$ and communication complexity of $O(\frac{n^{\frac{1}{r(n)}} \cdot r(n) \cdot log(\frac{r(n)}{\varepsilon})}{\varepsilon})$. In addition, the time complxity of the dsIPP is $T_P = Q_P$ and $T_V = O(\frac{n^{\frac{1}{r(n)}} \cdot r(n) \cdot log(\frac{r(n)}{\varepsilon})}{\varepsilon})$.*

4 Tolerant dsIPP for ROOBPs of Constant Width

In this section, we present a tolerant dsIPP for ROOBPs of constant width. Before we review our plan in Section 4.2, let us recall some standard definitions related to ROOBPs.

4.1 Definitions

▶ **Definition 11** (ROOBP). *A **branching program (BP)** on n variables is a directed acyclic graph that has a unique source vertex (denoted s) with in-degree 0 and a unique sink vertex (denoted t) with out-degree 0. Each non-sink vertex is labeled by an index $i \in [n]$, and has 2 outgoing edges, which are labeled by either 0 or 1. An input $x \in \{0,1\}^n$ defines a walk on B starting at the source vertex, such that at every vertex labeled by $i \in [n]$, the step taken is on the edge labeled by x_i. The output of the branching program B on input $x \in \{0,1\}^n$, denoted $B(x)$, is defined as 1 if the walk reached the sink vertex, and 0 if it got "stuck" before reaching it (i.e., the walk reached vertex labeled by $i \in [n]$, and there was no outgoing edge labeled by x_i).*

*An **oblivious BP** is a BP in which the nodes are partitioned into levels, $L_0, ..., L_n$, and edges are going only from one level to nodes in the consecutive level. In addition, all the vertices of some level are associated with the same index. Therefore, we can say that the level itself is associated with some index. A **read-once oblivious BP (ROOBP)** is a BP in which no two levels are associated with the same index. An **ROOBP of constant width** w is an ROOBP in which every level has at most w vertices.*

From now on, let B be an ROOBP with n variables and width at most w.

▶ Remark 12. By the above, there's a 1-1 correspondence between an $s \rightsquigarrow t$ path in B and an **accepting** input for B. In that case, we say that the $s \rightsquigarrow t$ path is **associated** with the accepting input.

▶ Remark 13. We say that two vertices $u \in L_l$ and $v \in L_{l'}$ in B are connected only if they are connected via a **directed** path. If $l < l'$, we say that u is **forwards-connected** to v, and v is **backwards-connected** to u.

▶ Remark 14. Without loss of generality, we assume:

1. For every $i \in \{0, ..., n-1\}$, L_i is associated with the index i. Otherwise, we can change the input x to an input x' by reordering its indices accordingly.
2. B depends on the entire input. Otherwise, we can change the input to $x' \in \{0,1\}^{n'}$, and then B will depend on every index of x'.
3. There exists a path between the source and the sink of B; this is equivalent to assuming there exists an $x \in \{0,1\}^n$ accepted by B.

▶ **Definition 15** (Absolute and Relative Distance). *We denote the **absolute distance** between two strings $x, x' \in \{0,1\}^n$ by $\Delta(x, x') = |\{x[i] \neq x'[i] : i \in [n]|$, and their **relative distance** by $\overline{\Delta}(x, x') = \frac{\Delta(x,x')}{n}$. Assume there exist an accepting input for B, an ROOBP of length n.*

We denote the ***absolute distance*** *of a string $x \in \{0,1\}^n$ from B by $\Delta(x,B) = min\{\Delta(x,x') : x' \in \{0,1\}^n, B(x') = 1\}$. Similarly, the* ***relative distance*** *of x from B is denoted by $\overline{\Delta}(x,B) = min\{\overline{\Delta}(x,x') : x' \in \{0,1\}^n, B(x') = 1\}$.*

4.2 Our Plan

For a fixed ROOBP B of width w, proximity parameters $\varepsilon_f > \varepsilon_c \geq 0$, and oracle access to an input $x \in \{0,1\}^n$, we present an r-round IPP, with $r = \log(n)$ and with poly-logarithmic communication complexity, in which V uses query complexity of $O(\frac{1}{\varepsilon_f - \varepsilon_c})$, the honest prover P uses poly-logarithmic query complexity, and the following holds:

- If $\overline{\Delta}(B,x) \leq \varepsilon_c$, then when communicating with P, with probability at least $2/3$ the verifier V accepts x.
- If $\overline{\Delta}(B,x) > \varepsilon_f$, then when communicating with any prover P^*, with probability at least $2/3$ the verifier V rejects x.

We note that we use r to represent the number of (pairs) of message exchanges (and not the total number of messages sent).

Our plan is as follows: First, in Section 4.3, we present a standard tolerant IPP for the ROOBP Problem. This IPP (which is not doubly sub-linear) uses a divide-and-conquer approach, and in it the prover P partitions the ROOBP to a sequence of k sub-ROOBPs (see Definition 16) according to the accepting path associated with an accepting input that minimizes the distance to x in B (see Definition 17). Then, in Section 4.4, we present a warm-up towards the tolerant dsIPP: We amend the standard tolerant IPP by allowing P to partition the ROOBP according to an approximated path, and relaxing the accepting conditions of V in a way that doesn't compromise the completeness and soundness of the protocol. Unfortunately, in this IPP the prover P still has query complexity of $\Omega(n)$. Lastly, in Section 4.5, we present the actual tolerant dsIPP, which applies the IPP presented in Section 4.4 recursively, thus improving the query complexity of P without compromising the query complexity of V.

4.3 The Standard Tolerant IPP

Given some ROOBP B, and proximity parameters $\varepsilon_f > \varepsilon_c \geq 0$, we want to verify that $\overline{\Delta}(B,x) \leq \varepsilon_c$. In the following tolerant IPP, that adapts the non-tolerant IPP from [7], we decompose B to k consecutive equally-length sub-ROOBPs, $B_1, ..., B_k$. Therefore, we start with defining the notion of a sub-ROOBP.

▶ **Definition 16** (sub-ROOBP). *For $0 \leq i \leq j \leq n$, let $u \in L_i$ and $v \in L_j$. We define $B[u,v]$ as a* ***sub-ROOBP*** *of B, with the following properties:*

- *The source (resp., sink) vertex of $B[u,v]$ is u (resp., v).*
- *$B[u,v]$ is of length $j - i$.*
- *If $x \in \{0,1\}^n$ is the input for B, then $x[i+1,j] = x[i+1] \cdot x[i+2] \cdot \cdots x[j]$ is the input for $B[u,v]$.*

Indeed, there is more than one way we can decompose B to k consecutive equally-length sub-ROOBPs, because for each sub-ROOBP there could be up to w^2 possible source-sink pairs. Therefore, we continue with defining the notion of (what we consider) a valid decomposition of B to k sub-ROOBPs: A decomposition that doesn't cause us to "lose" any distance (i.e. the average distance of x from all the sub-ROOBPs in the decomposition is at least the distance of x from B). We can achieve this by enforcing the decomposition to follow some path associated with some accepting input for B, since for any accepting input x' we have $\overline{\Delta}(x,x') \geq \overline{\Delta}(x,B)$.

▶ **Definition 17** ((k,π)-Decomposition). *Let B be an ROOBP, and let π be an accepting path in B. The **(k,π)-decomposition** of B is a sequence of k sub-ROOBPs of B, denoted $\mathcal{B}_{k,\pi} = (B_1, ..., B_k)$, such that for every $i \in [k]$, the source of B_i is the $\frac{(i-1)\cdot n}{k}$'th vertex in π, and the sink of B_i is the $\frac{i\cdot n}{k}$'th vertex in π.*

Now, we observe that when we partition x to k consecutive equally-length parts, $x_1, ..., x_k$, the following holds:

- On the one hand, if x is ε_c-close to B, then for $(B_1, ..., B_k)$, the (k,π)-decomposition of B constructed according to an accepting input that minimizes the distance to x, it holds that $\sum_i \frac{\overline{\Delta}(B_i,x)}{k} \leq \varepsilon_c$.
- On the other hand, if $\overline{\Delta}(B,x) > \varepsilon_f$, then for any $(B_1, ..., B_k)$, a (k,π)-decomposition of B constructed according to some accepting path in B, it holds that $\sum_i \frac{\overline{\Delta}(B_i,x_i)}{k} > \varepsilon_f$.

Given the above, we consider the following IPP: The prover P finds the (k,π)-decomposition of B constructed according to an accepting input that minimizes the distance to x. Then, P checks the distance of x_i from each of the sub-ROOBPs in the decomposition, and sends V a succinct representation of the decomposition (for example, the sink of every sub-ROOBP), as well as the obtained distances. After V receives some decomposition $(B_1, ..., B_k)$ and claimed distances $(\hat{\varepsilon_1}, ..., \hat{\varepsilon_k})$, V verifies that $(B_1, ..., B_k)$ is a valid (k,π)-decomposition of B, and that the average of the claimed distances is at most ε_c. Lastly, V selects at random $i \in [k]$, and verifies that $\overline{\Delta}(B_i, x_i) = \hat{\varepsilon}_i$ (by reading the entire x_i).

Note that if x is ε_c-close to B and V interacts with P, then $\sum_i \frac{\hat{\varepsilon}_i}{k} \leq \varepsilon_c$, and for every $i \in [k]$, it holds that $\overline{\Delta}(B_i, x_i) = \hat{\varepsilon}_i$, and the verifier always accepts. However, if $\overline{\Delta}(B,x) > \varepsilon_f$, then for any valid decomposition $(B_1, ..., B_k)$ and claimed distances $(\hat{\varepsilon}_1, ..., \hat{\varepsilon}_k)$ sent by some prover P^* such that $\sum_i \frac{\hat{\varepsilon}_i}{k} \leq \varepsilon_c$, it holds that $\sum_i \frac{\overline{\Delta}(B_i,x_i)-\hat{\varepsilon}_i}{k} > \varepsilon_f - \varepsilon_c$. Then, the average deviation of the actual distance of x from B, from the claimed distance of x from B, translates to a deviation on a corresponding fraction of random i's, which guarantees that the verifier rejects with probability at least $\varepsilon_f - \varepsilon_c$. To increase the rejection probability, we can repeat this procedure for $O(\frac{1}{\varepsilon_f - \varepsilon_c})$ times.

We stress that V can verify the validity of a (k,π)-decomposition without any query access to the input x. Thus, the query complexity of the verifier is the length of a single x_i times the number of repetitions, which gives $O(\frac{n}{(\varepsilon_f - \varepsilon_c)\cdot k})$. However, to find an accepting input that minimizes the distance to x, the prover must access to the entire input x, which yields query complexity of $\Omega(n)$.

4.4 A Warm-Up towards a tolerant dsIPP

In this section, we aim to achieve poly-logarithmic query complexity for the honest prover P without compromising the query complexity of the verifier V: We amend the interaction described in Section 4.3 by allowing P to divide the ROOBP B according to an accepting input that *approximates* the distance of x to B. However, at the end of this section, we explain why P still has query complexity of $\Omega(n)$. In Section 4.5, we show how to solve this issue and present the actual tolerant dsIPP.

In the full version of this paper, we show the existence of a δ-distance-approximation algorithm for ROOBPs of constant width with query complexity that is independent of the length of the input, n. That is, we show there exists an algorithm that given an ROOBP B of width w, an input $x \in \{0,1\}^n$, deviation parameter $\delta > 0$ and error probability parameter $\eta > 0$, outputs $\hat{\varepsilon}$ such that with probability at least $1 - \eta$ it holds that $|\hat{\varepsilon} - \overline{\Delta}(B,x)| \leq \delta$, and this algorithms has query complexity of $(\frac{\log(\frac{1}{\eta})\cdot 2^w}{\delta})^{O(w)}$.

Based on this distance-approximation algorithm, we amend the interaction as follows: P performs distance approximation (with parameters δ and η we fix later) for every (k,π)-decomposition (constructed according to every accepting path π). Since P performs distance approximation also for the decomposition of B constructed according to an accepting input that minimizes the distance to x, P can just choose the the (k,π)-decomposition which yields the minimal (average) distance approximation. That is, P chooses $\mathcal{B}_{k,\pi^*} = (B_1^{\pi^*},...,B_k^{\pi^*})$ for which $\sum_i \frac{\hat{\varepsilon}_i^{\pi^*}}{k}$ is minimal.

Even though there could be many possible (k,π)-decompositions, the total number of sub-ROOBPs in all the (k,π)-decompositions is upper-bounded by $w^2 \cdot k$: For every $i \in [k]$, there are at most w^2 possible source-sink pairs for the i't sub-ROOBP. Thus, using the union bound, with probability at least $1 - w^2 \cdot k \cdot \eta$, all approximations deviate from the actual distance by at most δ, and the following holds:

- For every $i \in [k]$, it holds that $\overline{\Delta}(B_i^{\pi^*}, x_i) \leq \hat{\varepsilon}_i^{\pi^*} + \delta$.
- If $\overline{\Delta}(B,x) \leq \varepsilon_c$, then since P chooses the (k,π)-decomposition which yields the minimal (average) distance approximation, we get that $\sum_i \frac{\hat{\varepsilon}_i^{\pi^*}}{k} \leq \varepsilon_c + \delta$.

Lastly, we also change V's accepting conditions in the interaction, so that V allows the deviation. That is, after V receives $(B_1,...,B_k)$ and $(\hat{\varepsilon}_1,...,\hat{\varepsilon}_k)$, in addition to verifying that $(B_1,...,B_k)$ is a valid decomposition, V verifies that $\sum_i \frac{\hat{\varepsilon}_i}{k} \leq \varepsilon_c + \delta$, and after V chooses $i \in [k]$ uniformly at random, V verifies that $\overline{\Delta}(B_i, x_i) \leq \hat{\varepsilon}_i + \delta$ (by reading the entire x_i). Similar to the standard IPP, we repeat the protocol for $O(\frac{1}{\varepsilon_f - \varepsilon_c})$ times.

4.4.1 Correctness

We claim that the completeness of the protocol still holds (but with a completeness error). If V interacts with P, then for a single invocation of the protocol, using the union bound, with probability at least $1 - \eta \cdot k \cdot w^2$, all approximations deviate from the actual distances by at most δ. Then, if $\overline{\Delta}(B,x) \leq \varepsilon_c$, it holds that $\sum_i \frac{\hat{\varepsilon}_i^{\pi^*}}{k} \leq \varepsilon_c + \delta$, and for every $i \in [k]$ it holds that $\overline{\Delta}(B_i^{\pi^*}, x_i) \leq \hat{\varepsilon}_i^{\pi^*} + \delta$, and the verifier accepts. Since we invoke the protocol for $O(\frac{1}{\varepsilon_f - \varepsilon_c})$ times, then for $t = O(\frac{k \cdot w^2}{\varepsilon_f - \varepsilon_c})$, the verifier accepts in all invocations with probability at least $1 - \eta \cdot t$. Thus, we can set $\eta \leq 0.01 \cdot \frac{1}{t} = O(\frac{\varepsilon_f - \varepsilon_c}{k \cdot w^2})$, and the verifier accepts with high constant probability in all invocations.

To show the soundness of the protocol also holds, assume $\overline{\Delta}(B,x) > \varepsilon_f$, and let $(B_1,...,B_k)$ be a decomposition that P^* sends, along with the respective claimed distance approximations $(\hat{\varepsilon}_1,...,\hat{\varepsilon}_k)$. If the decomposition is valid, then $\sum_i \frac{\overline{\Delta}(B_i,x_i)}{k} > \varepsilon_f$. Thus, if $\sum_i \frac{\hat{\varepsilon}_i}{k} \leq \varepsilon_c + \delta$, we get that $\sum_i \frac{\overline{\Delta}(B_i,x_i) - \hat{\varepsilon}_i}{k} > \varepsilon_f - \varepsilon_c - \delta$. Then, the average deviation of the actual distance of x from B, from the claimed distance of x from B, beyond the allowed deviation, translates to a deviation on a corresponding fraction of random i's. Since the verifier rejects if $\overline{\Delta}(B_i,x_i) > \hat{\varepsilon}_i + \delta$, we can set $\delta = \frac{\varepsilon_f - \varepsilon_c}{3}$, and infer that the verifier still rejects with probability at least $\Omega(\varepsilon_f - \varepsilon_c)$ in a single invocation. Since we invoke the protocol for $O(\frac{1}{\varepsilon_f - \varepsilon_c})$ times, we get that the verifier rejects with high constant probability in at least one of the invocations.

4.4.2 Query Complexity

Now, let us analyze the query complexity of V and P:

- Because V can verify that the (k,π)-decomposition that P sends is valid without any query access to the input x, the query complexity of V is $O(\frac{n}{(\varepsilon_f - \varepsilon_c) \cdot k})$: V reads a sub-sequence of the input of length $\frac{n}{k}$ for $O(\frac{1}{\varepsilon_f - \varepsilon_c})$ repetitions.

- The query complexity of P is $k \cdot (\frac{\log(\frac{k}{\varepsilon_f - \varepsilon_c}) \cdot 2^w}{\varepsilon_f - \varepsilon_c})^{O(w)}$, because the distance approximation algorithm with deviation parameter $\delta = O(\varepsilon_f - \varepsilon_c)$ and error probability parameter $\eta = O(\frac{\varepsilon_f - \varepsilon_c}{w^2 \cdot k})$ has query complexity of $(\frac{\log(\frac{k}{\varepsilon_f - \varepsilon_c}) \cdot 2^w}{\varepsilon_f - \varepsilon_c})^{O(w)}$, and P invokes the algorithm for $w^2 \cdot k$ times in each of the $O(\frac{1}{\varepsilon_f - \varepsilon_c})$ repetitions.

Since we can δ-approximate $\overline{\Delta}(B, x)$ without any interaction using query complexity $(\frac{2^w}{\delta})^{O(w)}$, we can perform tolerant $(\varepsilon_f, \varepsilon_c)$-test for ROOBPs using query complexity $(\frac{2^w}{\varepsilon_f - \varepsilon_c})^{O(w)}$. Therefore, for the IPP to be meaningful, we want the query complexity of V to be $o((\frac{2^w}{\varepsilon_f - \varepsilon_c})^{O(w)})$. To achieve this in our setting, we must require $k = \omega((\frac{\varepsilon_f - \varepsilon_c}{2^w})^{O(w)}) \cdot n)$. However, this causes the query complexity of P to be $\Omega(n)$, whereas we want our IPP to be doubly sub-linear. In the next section, we extend the foregoing IPP by applying it recursively, and show that we can get both query complexity of $o((\frac{2^w}{\varepsilon_f - \varepsilon_c})^{O(w)})$ for V, **and** poly-logarithmic query complexity for P.

4.5 The Actual dsIPP

In Section 4.4, we aim to achieve poly-logarithmic query complexity for the honest prover P, without compromising the query complexity of the verifier V. Unfortunately, for reasons explained at Section 4.4.2, the prover P still has query complexity of $\Omega(n)$.

To solve this, we extend the IPP presented in Section 4.4 by applying it recursively: That is, after V selects at random $i \in [k]$, instead of directly checking the distance of x_i from B_i, both parties recursively run the protocol on x_i with B_i, and do it for r rounds. Only in the base of the recursion, assuming the input is x' and the ROOBP is B', V checks the distance of x' from B' and accepts accordingly. This gives us an improvement in the query complexity of the verifier, as the length of the input after r rounds is $(\frac{n}{k^r})$, while the number of times the prover needs to run the distance approximation algorithm grows only to $w^2 \cdot k \cdot r$ (in every repetition). In order to make the query complexity of V independent of the input length, we set $k = n^{\frac{1}{r}}$.

Now, we observe the following: Assume V interacts with P, with input x and ROOBP B, such that $\overline{\Delta}(B, x) \leq \varepsilon_c$. Then, since we are only guaranteed that $\sum_i \frac{\hat{\varepsilon}_i}{k} \leq \varepsilon_c + \delta$, when both parties recursively apply the protocol with input x_i and ROOBP B_i, we can't guarantee that $\overline{\Delta}(B_i, x_i) \leq \varepsilon_c$ (or even guarantee that $\overline{\Delta}(B_i, x_i) \leq \varepsilon_c + \delta$).

Therefore, we amend the protocol as follows: In addition to the input x and the ROOBP B, we add a claimed distance parameter, $\hat{\varepsilon}$, that represents P's claim regarding the distance of x from B. In the first recursion level, we set $\hat{\varepsilon} = \varepsilon_c$, since P claims that $\overline{\Delta}(B, x) \leq \varepsilon_c$. Then, at each recursion level, assume our input x', our ROOBP is B', our (new) claimed distance parameter is $\hat{\varepsilon}'$, and P sends the decomposition $(B_1, ..., B_k)$ and the respective approximations $(\hat{\varepsilon}'_1, ..., \hat{\varepsilon}'_k)$. Then, V verifies that $(B_1, ..., B_k)$ is a valid decomposition and that $\sum_i \frac{\hat{\varepsilon}'_i}{k} \leq \hat{\varepsilon}' + \delta$ (i.e., we replace ε_c in the previous condition with the new parameter $\hat{\varepsilon}'$). Lastly, for some $i \in [k]$ both parties recursively run the protocol on x'_i with B'_i and the claimed distance $\hat{\varepsilon}'_i$, and at the base of the recursion, V verifies that the distance of the input from the ROOBP does not exceed the **claimed distance** by more than δ.

Now, on every recursion level, P only claims that the average distance doesn't grow too much **with respect to the approximated distance in the previous recursion level**. Since P uses a distance approximation algorithm with some error probability parameter $\eta > 0$, and chooses the (k, π)-decomposition of B that yields the minimal average distance approximation from x, the claim holds for all recursion levels with probability at least $1 - \eta \cdot k \cdot w^2 \cdot r_0$, and at the base level, the verifier accepts. Since P invokes the protocol for

$O(\frac{1}{\varepsilon_f-\varepsilon_c})$ times, then for $t = O(\frac{w^2\cdot k\cdot r}{\varepsilon_f-\varepsilon_c})$, the verifier accepts in all invocations with probability at least $1-\eta\cdot t$. Thus, we can set $\eta = 0.01\cdot\frac{1}{t} = O(\frac{\varepsilon_f-\varepsilon_c}{w^2\cdot k\cdot r})$, and the verifier accepts with high constant probability in all invocations. Lastly, we note that at each recursion level, we want V to account for two possible deviations: In the approximation that P provides for the previous recursion level (i.e. $\hat{\varepsilon}$), and in the approximations that P performs in the current recursion level (i.e. $(\hat{\varepsilon}_1,...,\hat{\varepsilon}_k)$).

For soundness, we observe that V allows the prover to exceed from the initial claimed distance (i.e., ε_c) by at most δ for r times: At each of the recursion levels, as well as in the base level. Thus, to make sure that soundness still holds (i.e. V does not allow the prover to exceed too much in distance), we set δ to be smaller (by a factor of r). Details follow.

▶ **Protocol 2.** *Tolerant dsIPP for the ROOBP Problem*

- ***Query Access Input:*** $x\in\{0,1\}^n$
- ***Proximity Parameters:*** $\varepsilon_f > \varepsilon_c \geq 0$
- ***Massive Parameter:*** *ROOBP B of width w*
- ***Other Parameters:*** *Claimed distance $\hat{\varepsilon}$ (initialized to ε_c in the first round), initial number of rounds r_0 and remaining number of rounds r*

1. *Set $k = n^{\frac{1}{r_0}}$, $\delta = \frac{\varepsilon_f-\varepsilon_c}{3r_0}$, and $\eta = O(\frac{\varepsilon_f-\varepsilon_c}{r_0\cdot k\cdot w^2})$.*
2. *V: If $r=0$ then accept iff $\overline{\Delta}(B,x)\leq\hat{\varepsilon}+\frac{\delta}{2}$.*
3. *P:*
   a. *For every $i\in[k]$, and for every $u\in L_{\frac{(i-1)\cdot n}{k}}$ and $v\in L_{\frac{i\cdot n}{k}}$, approximate the distance of x_i from $B[u,v]$ with deviation parameter $\frac{\delta}{2}$ and error probability parameter η.*
   b. *For every (k,π)-decomposition of B, $\mathcal{B}_{k,\pi} = (B_1^\pi,...,B_k^\pi)$, let $(\hat{\varepsilon}_1^\pi,...,\hat{\varepsilon}_k^\pi)$ be the respective distance approximations obtained in the previous step.*
   c. *Find a decomposition $\mathcal{B}_{k,\pi^*}$ for which $\sum_i\hat{\varepsilon}_i^{\pi^*}$ is minimal, and send (a succinct representation of) $\mathcal{B}_{k,\pi^*}$, along with $(\hat{\varepsilon}_1^{\pi^*},...,\hat{\varepsilon}_k^{\pi^*})$, to V.*
4. *V:*
   a. *Verify that $\mathcal{B}_{k,\pi^*}$ is a valid (k,π)-decomposition, and that $\sum_i\frac{\hat{\varepsilon}_i^{\pi^*}}{k}\leq\hat{\varepsilon}+\delta$. Otherwise, reject.*
   b. *Choose uniformly at random $i\in[k]$ and send i to P.*
5. *Both parties recursively invoke the protocol with input x_i, ROOBP $B_i^{\pi^*}$, claimed distance $\hat{\varepsilon}_i^{\pi^*}$, and remaining number of rounds $r-1$.*

Initiate the protocol with $\hat{\varepsilon}=\varepsilon_c$, repeat the protocol for $O(\frac{1}{\varepsilon_f-\varepsilon_c})$ times, and accept iff V accepts in all repetitions.

▶ Remark 18. P can efficiently find a decomposition in Step 3(c) as follows: Construct a graph $G=(V,E)$ with V being all the vertices of B in the levels $(L_0,L_{\frac{n}{k}},L_{\frac{2n}{k}}...,L_{\frac{k\cdot n}{k}})$, and $u,v\in V$ are connected in G with edge-weight $\hat{\varepsilon}_{u,v}$ if P performs distance approximation for $B[u,v]$ in Step 3(a) and it results in $\hat{\varepsilon}_{u,v}$. Then, every (k,π)-decomposition considered in Step 3(b) has a corresponding $s\rightsquigarrow t$ path in G, and the weight of the $s\rightsquigarrow t$ path equals the sum of distance approximations for the sub-ROOBPs in the decomposition. Thus, P can choose the decomposition with the shortest $s\rightsquigarrow t$ path in G.

We show that Protocol 2 is a dsIPP for ROOBPs: In Section 4.5.1, we show the correctness of the protocol, and in Section 4.6.1, we present the query complexity of the protocol.

4.5.1 Correctness

Completeness. Assume $\overline{\Delta}(B,x) \leq \varepsilon_c$ and V interacts with P. We observe that at any recursion level (but the base level), the following claim holds: If our input is x', our ROOBP is B' and our claimed distance parameter is $\hat{\varepsilon}'$ such that $\overline{\Delta}(B',x') \leq \hat{\varepsilon}' + \frac{\delta}{2}$, then with probability at least $1 - \eta \cdot k \cdot w^2$, the prover P sends $(B'_1, ..., B'_k)$ and $(\hat{\varepsilon}'_1, ..., \hat{\varepsilon}'_k)$ to V such that the following holds:

1. The verifier doesn't reject in Step $4(a)$ (i.e., $\sum_i \frac{\hat{\varepsilon}'_i}{k} \leq \hat{\varepsilon}' + \delta$).
2. The protocol is recursively invoked with some x'_i, B'_i and $\hat{\varepsilon}'_i$ for which $\overline{\Delta}(B'_i, x'_i) \leq \hat{\varepsilon}'_i + \frac{\delta}{2}$.

The claims follows from the fact that P runs the distance approximation algorithm with deviation parameter $\frac{\delta}{2}$ and with error probability parameter η. Since we start with $\overline{\Delta}(B,x) \leq \varepsilon_c = \hat{\varepsilon}$, then with probability at least $1 - \eta \cdot k \cdot w^2 \cdot r_0$, the verifier doesn't reject in Step $4(a)$ in all the r_0 recursion levels. By the second item of the claim, the base of the recursion is also invoked with an input x', ROOBP B' and claimed distance parameter $\hat{\varepsilon}'$ such that $\overline{\Delta}(B',x') \leq \hat{\varepsilon}' + \frac{\delta}{2}$, and the verifier accepts. Since we repeat the protocol for $O(\frac{1}{\varepsilon_f - \varepsilon_c})$ times, we get that for $t = O(\frac{k \cdot w^2 \cdot r_0}{\varepsilon_f - \varepsilon_c})$, the verifier accepts in all invocations with probability at least $1 - \eta \cdot t$. Then, since we set $\eta = 0.01 \cdot \frac{1}{t} = O(\frac{\varepsilon_f - \varepsilon_c}{k \cdot w^2 \cdot r_0})$, we get that V accepts with high constant probability in all invocations.

Soundness. Assume $\overline{\Delta}(B,x) > \varepsilon_f$. We observe that in each round, we lose at most an additive factor of δ in distance (i.e. we allow the prover to "exceed" the claimed distance up to an additive δ factor). Since we start with claimed distance ε_c, and since we set $\delta = \frac{\varepsilon_f - \varepsilon_c}{3r_0}$, we get that after r_0 rounds, we are still at distance at least $\frac{\varepsilon_f - \varepsilon_c}{2}$ from a valid assertion. Hence, the verifier rejects with probability at least $\Omega(\varepsilon_f - \varepsilon_c)$ *in each invocation*, and rejects with high constant probability in at least one of the $O(\frac{1}{\varepsilon_f - \varepsilon_c})$ invocations.

4.6 Computational Complexity (of Protocol 2)

The query and communication complexity of our tolerant dsIPP are our main complexity measures. Indeed, the communication complexity is $O(\frac{k \cdot \log(w \cdot n) \cdot r_0}{\varepsilon_f - \varepsilon_c})$, because in each of the $O(\frac{1}{\varepsilon_f - \varepsilon_c})$ repetitions there are r_0 rounds, and in each round the prover P sends k vertices and k distance approximations to V that can be represented by $O(k \cdot \log(w \cdot n))$ bits. Since we set $k = n^{\frac{1}{r_0}}$, we can set $r_0 = \log(n)$ and get poly-logarithmic communication complexity.

We continue in Section 4.6.1, where we calculate the query complexity of both the verifier and the prover. We show that if we set $r_0 = \log(n)$, then V uses query complexity of $O(\frac{1}{\varepsilon_f - \varepsilon_c})$, and the honest prover P uses poly-logarithmic query complexity.

We finish by calculating the time complexity of our tolerant dsIPP. We show that, if we set $r_0 = \log(n)$, and assume both V and P have black-box access to two procedures that refer to the input ROOBP (see Section 4.6.2 for more details), then the verifier V has time complexity of $O(\frac{\log^2(n)}{\varepsilon_f - \varepsilon_c})$. In addition, if the honest prover P uses a distance approximation algorithm for ROOBPs with time complexity T_A, then P has time complexity of $O(\frac{\log^2(n) \cdot \mathsf{T}_A}{\varepsilon_f - \varepsilon_c})$.

4.6.1 Query Complexity

Let us analyze the query complexity of V (denoted Q_V) and P (denoted Q_P):

- Because V can verify that the (k,π)-decomposition P sends in every round is valid without any query access to the input x, we have $\mathsf{Q}_V = O(\frac{n}{(\varepsilon_f - \varepsilon_c) \cdot k^{r_0}}) = O(\frac{1}{\varepsilon_f - \varepsilon_c})$: V reads a sub-sequence of the input of length $\frac{n}{k^{r_0}} = 1$ for $O(\frac{1}{\varepsilon_f - \varepsilon_c})$ times.

- P uses a distance approximation algorithm A with deviation parameter $\delta = O(\frac{\varepsilon_f - \varepsilon_c}{r_0})$ and error probability parameter $\eta = O(\frac{\varepsilon_f - \varepsilon_c}{r_0 \cdot k \cdot w^2})$, with query complexity $\mathsf{Q}_A(\delta, \eta, w)$. Then, the query complexity of Q_P is $\mathsf{Q}_A(\delta, \eta, w)$ times the number of calls P makes to A in all the repetitions. Since P calls the approximator for $w^2 \cdot k \cdot r_0$ times in each of the $O(\frac{1}{\varepsilon_f - \varepsilon_c})$ repetitions, and we set $k = n^{\frac{1}{r_0}}$, we get that $\mathsf{Q}_P = O(\frac{w^2 \cdot n^{\frac{1}{r_0}} \cdot r_0}{\varepsilon_f - \varepsilon_c} \cdot \mathsf{Q}_A(\delta, \eta, w))$.

Hence, regardless of how we set r_0, the query complexity of the verifier is $O(\frac{1}{\varepsilon_f - \varepsilon_c})$. Indeed, the larger we set r_0, the smaller Q_P gets, but this causes the round complexity to grow. To minimize Q_P, we can set $r_0 = \log(n)$, and we get $\mathsf{Q}_P = O(\frac{w^2 \cdot \log(n)}{\varepsilon_f - \varepsilon_c} \cdot \mathsf{Q}_A(\delta, \eta, w))$.

In the full version of this paper, we show the existence of a distance approximation algorithm with query complexity of $\mathsf{Q}_A(\delta, \eta, w) = (\frac{\log(\frac{1}{\eta}) \cdot 2^w}{\delta})^{O(w)}$. If P uses this distance approximation algorithm, we get $\mathsf{Q}_P = \tilde{O}(n^{\frac{1}{r_0}}) \cdot (\frac{r_0 \cdot 2^w}{\varepsilon_f - \varepsilon_c})^{O(w)}$. Again, we can set $r_0 = \log(n)$, and if we assume w is constant, we get poly-logarithmic query complexity.

4.6.2 Time Complexity

Indeed, assuming a standard access to the input ROOBP B (i.e. the "Massive Parameter"), the time complexity of both the verifier (denoted T_V) and the honest prover (denoted T_P) is $\Omega(n)$:

- Given a sub-sequence of vertices $(v_0, ..., v_k)$ in B, the verifier V has to verify that every v_i is in the $\frac{i \cdot k}{n}$'th level of B, and that $v_i \rightsquigarrow v_{i+1}$. Since $v_0 \in L_0$ and $v_k \in L_n$, this requires accessing the entire ROOBP, thus yields time complexity of $\Omega(n)$.
- P approximates the distance of sub-parts of x from sub-ROOBPs of B of length $\frac{n}{k}$ for $w^2 \cdot k$ times. Using the distance approximation algorithm presented in the full version of this paper, this also yields time complexity of $\Omega(n)$.

However, we can take an alternative approach in the analysis: Assume both V and P have black-box access to the following two procedures, that refer to the structure of the input ROOBP, and are independent of the specific input x:

1. Given two vertices u and v, determine whether $u \rightsquigarrow v$ in B.
2. Given $i \in \{0, ..., n\}$, list the (up to w) vertices that are in level i of B.

In addition, assume P uses a distance approximation algorithm A with deviation parameter $\delta = O(\frac{\varepsilon_f - \varepsilon_c}{r_0})$ and error probability parameter $\eta = O(\frac{\varepsilon_f - \varepsilon_c}{r_0 \cdot k \cdot w^2})$, with time complexity $\mathsf{T}_A(\delta, \eta, n, w)$ (where n is the length of the input and w is the width of the ROOBP). Now, for $i \in [r_0]$, we can analyze the time complexity of V and P *in the i'th round of the protocol*:

- V calls the first two procedures for $O(k)$ times, obtains values that can be represented by $O(k \cdot \log(w \cdot n))$ bits, and performs simple calculations on the obtained values.
- P calls the first two procedures for k times, and calls the distance approximation algorithm for at most $w^2 \cdot k$ times on inputs of length $\frac{n}{k^i}$. By doing the foregoing, P obtains values that can be represented by $O(w^2 \cdot k \cdot \log(w \cdot n))$ bits, and performs simple calculations on the obtained values. This yields time complexity of $O(w^2 \cdot k \cdot (\log(w \cdot n) + \mathsf{T}_A(\delta, \eta, \frac{n}{k^i}, w)))$.

Since V and P perform the foregoing for every $i \in [r_0]$ in each of the $O(\frac{1}{\varepsilon_f - \varepsilon_c})$ invocations of the protocol, and since we set $k = n^{\frac{1}{r_0}}$, we get that:

- $\mathsf{T}_V = O(\frac{n^{\frac{1}{r_0}} \cdot r_0 \cdot \log(w \cdot n)}{\varepsilon_f - \varepsilon_c})$
- $\mathsf{T}_P = O(\frac{w^2 \cdot n^{\frac{1}{r_0}} \cdot (r_0 \cdot \log(w \cdot n) + \sum_{i=1}^{r_0} \mathsf{T}_A(\delta, \eta, n^{\frac{r_0 - i}{r_0}}, w))}{\varepsilon_f - \varepsilon_c})$

To minimize the time complexity, we can set $r_0 = \log(n)$, and assuming w is constant, we get $\mathsf{T}_V = O(\frac{\log^2(n)}{\varepsilon_f - \varepsilon_c})$ and $\mathsf{T}_P = O(\frac{\log^2(n)+\log(n)\cdot \mathsf{T}_A(\delta,\eta,n,w)}{\varepsilon_f - \varepsilon_c})$. We conclude with the following theorem and corollary.

▶ **Theorem 19.** *For every function $r : \mathbb{N} \to \mathbb{N}$, there exists an $r(n)$-round tolerant dsIPP for ROOBPs of width w with proximity parameters $\varepsilon_f > \varepsilon_c \geq 0$. In the tolerant dsIPP, the honest prover P uses a distance approximation algorithm A with deviation parameter $\delta = O(\frac{\varepsilon_f - \varepsilon_c}{r(n)})$ and error probability parameter $\eta = O(\frac{\varepsilon_f - \varepsilon_c}{r(n)\cdot n^{\frac{1}{r(n)}}\cdot w^2})$ such that the query complexity of A is $\mathsf{Q}_A(\delta,\eta)$ and the time complexity of A is $\mathsf{T}_A(\delta,\eta,n,w)$. Then, the computational complexity of the tolerant dsIPP is as follows:*

- ***Query Complexity**: $\mathsf{Q}_V = O(\frac{1}{\varepsilon_f - \varepsilon_c})$ and $\mathsf{Q}_P = O(\frac{w^2\cdot n^{\frac{1}{r(n)}}\cdot r(n)}{\varepsilon_f - \varepsilon_c} \cdot \mathsf{Q}_A(\delta,\eta))$.*
- ***Communication Complexity**: $O(\frac{n^{\frac{1}{r(n)}}\cdot r(n)\cdot log(w\cdot n)}{\varepsilon_f - \varepsilon_c})$.*
- ***Time Complexity**: If both V and P have black-box access to procedures that refer to the structure of the input ROOBP, then:*
  - $\mathsf{T}_V = O(\frac{n^{\frac{1}{r(n)}}\cdot r(n)\cdot log(w\cdot n)}{\varepsilon_f - \varepsilon_c})$
  - $\mathsf{T}_P = O(\frac{w^2\cdot n^{\frac{1}{r(n)}}\cdot (r(n)\cdot log(w\cdot n) + \sum_{i=1}^{r(n)} \mathsf{T}_A(\delta,\eta,n^{\frac{r(n)-i}{r(n)}},w))}{\varepsilon_f - \varepsilon_c})$

In the full version of this paper, we show the existence of a distance approximation algorithm with time complexity of $\mathsf{T}_A(\delta,\eta,n,w) = (\frac{\log(\frac{1}{\eta})\cdot 2^w}{\delta})^{O(w)}\cdot n$. If P uses this distance approximation algorithm, we get the following corollary.

▶ **Corollary 20.** *For every function $r : \mathbb{N} \to \mathbb{N}$, there exists an $r(n)$-round tolerant dsIPP for ROOBPs of width w with $\mathsf{Q}_V = O(\frac{1}{\varepsilon_f - \varepsilon_c})$, $\mathsf{Q}_P = \tilde{O}(n^{\frac{1}{r(n)}})\cdot(\frac{r(n)\cdot 2^w}{\varepsilon_f - \varepsilon_c})^{O(w)}$, and communication complexity of $O(\frac{n^{\frac{1}{r(n)}}\cdot r(n)\cdot log(w\cdot n)}{\varepsilon_f - \varepsilon_c})$. If both V and P have black-box access to procedures that refer to the structure of the input ROOBP, then $\mathsf{T}_V = O(\frac{n^{\frac{1}{r(n)}}\cdot r(n)\cdot log(w\cdot n)}{\varepsilon_f - \varepsilon_c})$ and $\mathsf{T}_P = (\frac{r(n)\cdot 2^w\cdot log(n)}{\varepsilon_f - \varepsilon_c})^{O(w)}\cdot n$.*

References

1. Ran Canetti, Guy Even, and Oded Goldreich. Lower bounds for sampling algorithms for estimating the average. *Inf. Process. Lett.*, 53(1):17–25, 1995. doi:10.1016/0020-0190(94)00171-T.
2. Alessandro Chiesa and Tom Gur. Proofs of proximity for distribution testing. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 53:1–53:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICS.ITCS.2018.53.
3. Funda Ergün, Ravi Kumar, and Ronitt Rubinfeld. Fast approximate probabilistically checkable proofs. *Inf. Comput.*, 189(2):135–159, 2004. doi:10.1016/J.IC.2003.09.005.
4. Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017. doi:10.1017/9781108135252.
5. Oded Goldreich. On doubly-efficient interactive proof systems. *Found. Trends Theor. Comput. Sci.*, 13(3):158–246, 2018. doi:10.1561/0400000084.
6. Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998. doi:10.1145/285055.285060.
7. Oded Goldreich, Tom Gur, and Ron D. Rothblum. Proofs of proximity for context-free languages and read-once branching programs. *Inf. Comput.*, 261:175–201, 2018. doi:10.1016/J.IC.2018.02.003.

8 Oded Goldreich and Dana Ron. A sublinear bipartiteness tester for bounded degree graphs. *Comb.*, 19(3):335–373, 1999. doi:10.1007/S004930050060.

9 Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002. doi:10.1007/S00453-001-0078-7.

10 Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: Interactive proofs for muggles. *J. ACM*, 62(4):27:1–27:64, 2015. doi:10.1145/2699436.

11 Shafi Goldwasser, Guy N. Rothblum, Jonathan Shafer, and Amir Yehudayoff. Interactive proofs for verifying machine learning. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 41:1–41:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.ITCS.2021.41.

12 Tom Gur, Mohammad Mahdi Jahanara, Mohammad Mahdi Khodabandeh, Ninad Rajgopal, Bahar Salamatian, and Igor Shinkar. On the power of interactive proofs for learning. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 1063–1070. ACM, 2024. doi:10.1145/3618260.3649784.

13 Tom Gur, Yang P. Liu, and Ron D. Rothblum. An exponential separation between MA and AM proofs of proximity. *Comput. Complex.*, 30(2):12, 2021. doi:10.1007/S00037-021-00212-3.

14 Tal Herman and Guy N. Rothblum. Verifying the unseen: interactive proofs for label-invariant distribution properties. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 – 24, 2022*, pages 1208–1219. ACM, 2022. doi:10.1145/3519935.3519987.

15 Tal Herman and Guy N. Rothblum. Doubley-efficient interactive proofs for distribution properties. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 743–751. IEEE, 2023. doi:10.1109/FOCS57990.2023.00049.

16 Ilan Newman. Testing membership in languages that have small width branching programs. *SIAM J. Comput.*, 31(5):1557–1570, 2002. doi:10.1137/S009753970038211X.

17 Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *J. Comput. Syst. Sci.*, 72(6):1012–1042, 2006. doi:10.1016/J.JCSS.2006.03.002.

18 Pavel Pudlák. A lower bound on complexity of branching programs (extended abstract). In Michal Chytil and Václav Koubek, editors, *Mathematical Foundations of Computer Science 1984, Praha, Czechoslovakia, September 3-7, 1984, Proceedings*, volume 176 of *Lecture Notes in Computer Science*, pages 480–489. Springer, 1984. doi:10.1007/BFB0030331.

19 Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. *SIAM J. Comput.*, 50(3), 2021. doi:10.1137/16M1096773.

20 Guy N. Rothblum and Ron D. Rothblum. Batch verification and proofs of proximity with polylog overhead. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography – 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part II*, volume 12551 of *Lecture Notes in Computer Science*, pages 108–138. Springer, 2020. doi:10.1007/978-3-030-64378-2_5.

21 Guy N. Rothblum, Salil P. Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 793–802. ACM, 2013. doi:10.1145/2488608.2488709.

22 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996. doi:10.1137/S0097539793255151.

23 Hadar Strauss. Emulating computationally sound public-coin ipps in the pre-coordinated model. *Electronic Colloquium on Computational Complexity (ECCC)*, TR24-131, 2024. URL: https://eccc.weizmann.ac.il/report/2024/131/.

Simultaneous Haar Indistinguishability with Applications to Unclonable Cryptography

Prabhanjan Ananth
University of California, Santa Barbara, CA, USA

Fatih Kaleoglu
University of California, Santa Barbara, CA, USA

Henry Yuen
Columbia University, New York, NY, USA

Abstract

We study a novel question about nonlocal quantum state discrimination: how well can non-communicating – but entangled – players distinguish between different distributions over quantum states? We call this task *simultaneous state indistinguishability*. Our main technical result is to show that the players cannot distinguish between each player receiving independently-chosen Haar random states versus all players receiving the same Haar random state.

We show that this question has implications to unclonable cryptography, which leverages the no-cloning principle to build cryptographic primitives that are classically impossible to achieve. Understanding the feasibility of unclonable encryption, one of the key unclonable primitives, satisfying indistinguishability security in the plain model has been a major open question in the area. So far, the existing constructions of unclonable encryption are either in the quantum random oracle model or are based on new conjectures.

We leverage our main result to present the first construction of unclonable encryption satisfying indistinguishability security, with quantum decryption keys, in the plain model. We also show other implications to single-decryptor encryption and leakage-resilient secret sharing. These applications present evidence that simultaneous Haar indistinguishability could be useful in quantum cryptography.

2012 ACM Subject Classification Theory of computation → Cryptographic protocols; Theory of computation → Quantum information theory

Keywords and phrases Quantum, Haar, unclonable encryption

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.7

Related Version This paper is based on the following technical report:
Full Version: `https://arxiv.org/abs/2405.10274` [8]

Funding *Prabhanjan Ananth*: the National Science Foundation Grant No. 2329938.
Fatih Kaleoglu: the National Science Foundation Grant No. 2329938.
Henry Yuen: AFOSR awards FA9550-21-1-0040 and FA9550-23-1-0363, NSF CAREER award CCF-2144219, NSF award CCF-2329939, and the Sloan Foundation.

Acknowledgements The authors would like to thank Ludovico Lami for answering many questions related to non-local state discrimination.

1 Introduction

Quantum state discrimination [40] is a foundational concept with applications to quantum information theory, learning theory and cryptography. In a state discrimination task, a party receives ρ_x from an ensemble $\{\rho_x\}_{x\in\mathcal{X}}$ and has to determine which state it received. A compelling variant of this problem is concerned with the multi-party setting where there are two or more parties and each party receives a disjoint subset of qubits of ρ_x. This multi-party variant of state discrimination has also garnered interest from quantum information-theorists


© Prabhanjan Ananth, Fatih Kaleoglu, and Henry Yuen;
licensed under Creative Commons License CC-BY 4.0

focused on the LOCC (local operations and classical communication) model and quantum data hiding [50, 14, 29, 25]. Understanding the multi-party state discrimination problem in turn sheds light on the difficulty of simulating global measurements using local measurements [52].

An important aspect to consider when formulating the multi-party state discrimination problem is the resources shared between the different parties. If we allow the parties to share entanglement and also communicate with each other then this is equivalent to the original state discrimination task (against a single party) due to teleportation. Thus, in order for the multi-party setting to be distinct from the single-party setting, we need to disallow either shared entanglement or classical communication. This results in two different settings:

- Parties without shared entanglement: In this setting, the parties are allowed to communicate using classical channels but they cannot share entanglement. The extensive research on quantum data hiding and LOCC [29, 32, 39, 46, 51, 15, 23, 24, 25, 36, 48] are mainly concerned with this setting.
- Parties with shared entanglement: In this setting, the parties are allowed to share entanglement but they cannot communicate. On the contrary, this setting is relatively unexplored with the notable exception being the recent works of [45, 30]. There are good reasons to study multi-party state discrimination with shared entanglement. Firstly, it can be viewed as a subclass of semi-quantum games [18], which are non-local games with quantum questions and classical answers. Secondly, it has connections to unclonable cryptography [54, 1], an emerging area in quantum cryptography, as discussed in [45].

Our Work. We focus on the setting when the parties share entanglement but are not allowed to communicate. We introduce a new concept called *simultaneous state indistinguishability* (SSI). In the two-party version of the problem, we have two parties (say, Bob and Charlie) who receive as input one of two bipartite states $\{\rho_0, \rho_1\}$ and they are supposed to distinguish. The first half of ρ_b is given to Bob and the second half is given to Charlie for $b \in \{0, 1\}$.

We say that ρ_0 and ρ_1 are ϵ-*simultaneous state indistinguishable* if the probability that Bob and Charlie can simultaneously distinguish is at most ϵ. That is, $d_{\mathsf{TV}}(x, x') \leq \epsilon$, where $d_{\mathsf{TV}}(\cdot, \cdot)$ denotes the total variation distance, $x = (x_B, x_C) \leftarrow (\text{Bob, Charlie})(\rho_0)$ denotes the random variable corresponding to the joint outputs of Bob and Charlie given input ρ_0, and similarly $x' = (x'_B, x'_C) \leftarrow (\text{Bob, Charlie})(\rho_1)$ denotes the random variable corresponding to their joint outputs when given input ρ_1.

This is related to a recent concept introduced by [45] who considered the unpredictability (search) version of this problem whereas we are interested in indistinguishability. Looking ahead, for applications, it turns out that the indistinguishability notion is more amenable to carrying out proofs (e.g. in the security of unclonable encryption) compared to the unpredictability definition considered by [45]. This is largely due to the fact that our notion is more compatible with the hybrid technique.

An interesting special case of simultaneous-state-indistinguishability is when Bob and Charlie either receive copies of the same state $|\psi\rangle$ drawn from some distribution $\mathcal{D}$ or receive i.i.d. samples from $\mathcal{D}$, i.e.

$$\rho_0 = \mathop{\mathbb{E}}_{|\psi\rangle\sim\mathcal{D}} \psi^{\otimes t} \otimes \psi^{\otimes t} \quad \text{vs.} \quad \rho_1 = \mathop{\mathbb{E}}_{|\psi_B\rangle,|\psi_C\rangle\sim\mathcal{D}} {\psi_B}^{\otimes t} \otimes \psi_C^{\otimes t}.$$

Observe that if Bob and Charlie were allowed to make global entangled measurements then they can indeed distinguish by performing swap tests. However, it is not clear these two situations are distinguishable using local measurements, even with preshared entanglement between Bob and Charlie.

We study simultaneous state indistinguishability in the case that each party receives (copies of) a state drawn from the *Haar measure*.[1] Specifically, we consider the following definition:

(d, t, ε)-Simultaneous Haar Indistinguishability: We say that (d, t, ε)-simultaneous Haar indistinguishability holds if any two non-communicating and entangled adversaries Bob and Charlie can distinguish the following distributions with probability at most ε:

- Bob and Charlie each receive t copies of $|\psi\rangle$, where $|\psi\rangle$ is a d-dimensional Haar state,
- Bob receives t copies of $|\psi_B\rangle$ and Charlie receives t copies of $|\psi_C\rangle$, where $|\psi_B\rangle, |\psi_C\rangle$ are i.i.d d-dimensional Haar states.

In the default setting, both Bob and Charlie each output 1 bit. We also consider the setting when they output multiple bits.

Variants of this problem have been studied in different contexts before. Independently, two works, namely, Harrow [38] and Chen, Cotler, Huang and Li [22] showed (for the case when $t = 1$) that Bob and Charlie fail, except with probability negligible in the dimension d, in the above distinguishing experiment as long as they don't share any entanglement. In fact, Harrow's result proves something stronger: the indistinguishability holds even if the two parties exchange classical information (i.e., LOCC setting). Both works discuss the applications of this problem to well studied topics such as multiparty data hiding, local purity testing and separations between quantum and classical memory. Neither of the works [38, 22] addresses the setting when Bob and Charlie can share an arbitrary amount of entanglement.

1.1 Our Results

1.1.1 Main Result

We show the following:

▶ **Theorem 1** (Informal). *For any $d, t \in \mathbb{N}$, (d, t, ε)-Simultaneous Haar indistinguishability holds for $\varepsilon = O\left(\frac{t^2}{\sqrt{d}}\right)$.*

Our result complements the works of [38, 22] by showing that it is not possible to distinguish i.i.d versus identical Haar states either using the entanglement resource or using classical communication.

In the case when $t = 1$, we show that $(d, 1, \epsilon)$-simultaneous Haar indistinguishability does not hold for $\epsilon = O\left(\frac{1}{d}\right)$, which suggests that the above bound cannot be improved significantly. Perhaps surprisingly, our attack even holds in the setting when Bob and Charlie do not share any entanglement. This further indicates that for the problem of simultaneous Haar indistinguishability, the gap between the optimal success probabilities in the entangled and the unentangled cases is small.

1.1.2 Applications

Besides being a natural problem, simultaneous Haar indistinguishability has applications to unclonable cryptography [54, 1, 55, 17, 9, 28, 27, 19]. This is an area of quantum cryptography that leverages the no-cloning principle of quantum mechanics to design cryptographic notions for tasks that are impossible to achieve classically.

[1] The Haar measure over states is the unique measure where for all fixed unitaries U, if $|\psi\rangle$ is Haar-distributed then so is $U|\psi\rangle$.

Unclonable Encryption: Unclonable encryption is an encryption scheme with quantum ciphertexts that are unclonable. It was first introduced by Broadbent and Lord [17] and is now considered a fundamental notion in unclonable cryptography. There are two security notions of unclonable encryption, namely search and indistinguishability security, studied in the literature. The search security stipulates that any cloning adversary[2] after receiving a ciphertext of randomly chosen message m should not be able to guess m except with probability negligible in $|m|$. In the challenge phase, the cloning adversary receives as input (k, k), where k is the decryption key. The indistinguishability security imposes a stronger guarantee that any cloning adversary after receiving encryption of m_b, for a randomly chosen bit b and adversarially chosen message pair (m_0, m_1), cannot predict b except with probability negligibly close to $\frac{1}{2}$.

While we have long known that search security is feasible [17], establishing indistinguishability security has remained an important and intriguing open problem. Unclonable encryption satisfying indistinguishability is achievable in the quantum random oracle model [6, 7] or in the plain model (i.e., without oracles) based on new unproven conjectures [3]. Various generic transformations are also known that convert one-time unclonable encryption to the public-key variant [5] or those that convert unclonable encryption for one-bit messages to multi-bit messages [41]. Given that unclonable-indistinguishable encryption has interesting applications to copy-protection [5, 26], it is important we completely settle its feasibility in the plain model. Perhaps embarassingly, we do not how to establish feasibility in the plain model even under strong assumptions such as indistinguishability obfuscation [10, 31]!

We consider the notion of unclonable encryption, where the decryption keys are allowed to be quantum keys. This only affects the challenge phase of the security experiment, where the cloning adversary now receives as input many copies[3] of the quantum decryption key.

We show the following:

▶ **Theorem 2** (Informal). *There is an unclonable encryption scheme, with quantum decryption keys, satisfying indistinguishability security in the plain model.*

Ours is the first work to show that unclonable-indistinguishable encryption exists in the plain model albeit with quantum decryption keys. Our relaxation (i.e. decryption keys being quantum states) is not only natural and interesting, it also respects the essence of unclonable encryption. For a detailed perspective on the above result, we refer the reader to the full version. We leave open the problem of achieving unclonable encryption with *classical* decryption keys to future works; we note that the prior works only mentioned the open problem of constructing unclonable encryption in the plain model without explicitly mentioning the need for classical decryption keys.

Moving Beyond Quantum Decryption Keys: A potential approach to transform this scheme into another scheme where the decryption keys are binary strings is to generate the decryption key to be an obfuscation of the setup algorithm that produces decryption keys. It is an intriguing problem to formalize the requirements of the underlying quantum obfuscation scheme. At the bare minimum, we require that the quantum obfuscation scheme satisfies the property that the obfuscated program can be represented as a binary string. While

[2] A cloning adversary is a tri-partite adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$. $\mathcal{A}$ receives as input an unclonable state and produces a bipartite state given to $\mathcal{B}$ and $\mathcal{C}$ who are not allowed to communicate. Then, in the challenge phase, the cloning adversary receives as input $(\mathsf{ch}_{\mathcal{B}}, \mathsf{ch}_{\mathcal{C}})$ and then gives $\mathsf{ch}_{\mathcal{B}}$ to $\mathcal{B}$ and $\mathsf{ch}_{\mathcal{C}}$ to $\mathcal{C}$. Finally, $\mathcal{B}$ and $\mathcal{C}$ output their respective answers. Refer to [7] for an abstract modeling of unclonable security notions.

[3] We consider a security notion where the adversary receives at most t copies of the quantum decryption key and t is fixed ahead of time.

achieving quantum obfuscation has been a difficult open problem [16, 12, 11], obfuscating special classes of quantum algorithms (those that capture the setup algorithm of Theorem 2) could be relatively more tractable.

Our scheme supports one-bit messages and is one-time secure. It is an interesting future direction to extend the works of [5] and [41] to generically achieve unclonable encryption with quantum decryption keys in the public-key setting and for longer messages.

Single-Decryptor Encryption: Single-decryptor encryption [33] is a sister notion of unclonable encryption, where instead of requiring the ciphertexts to be unclonable, we instead require the decryption keys to be unclonable. Constructions of single-decryptor encryption in different settings are known from a variety of assumptions [33, 27, 7, 43]. There are two important security notions considered in the literature. In the *independent* setting, in the challenge phase, the cloning adversary gets two independently generated ciphertexts while in the *identical* setting, it gets copies of the same ciphertext. All the known constructions of single-decryptor encryption [27, 7, 43] are in the independent setting and specifically, there are no known constructions in the identical setting. This should not be surprising in light of [33] who showed the equivalence between single-decryptor encryption with identical ciphertexts and unclonable encryption, which suggests the difficulty in achieving the identical ciphertext setting.

We prove the following.

▶ **Theorem 3** (Informal). *There is a single-decryptor encryption scheme, with quantum ciphertexts, satisfying identical indistinguishability security.*

Ours is the first work to demonstrate the feasibility of single-decryptor encryption in the identical-challenge setting, albeit with quantum ciphertexts.

In addition to its applications to unclonable cryptography, simultaneous Haar indistinguishability can be used to construct leakage-resilient secret sharing.

Leakage-Resilient Secret Sharing: Leakage-resilient cryptography [42] is an area of cryptography that is concerned with the goal of building cryptographic primitives that are resilient to side-channel attacks. We are interested in designing secret sharing schemes that are leakage resilient. In a leakage-resilient secret sharing scheme, a leakage function is applied on each share and we require the guarantee that all the leakages put together are not sufficient enough to compromise the security of the secret sharing scheme. Leakage-resilient secret sharing is a well studied topic [35, 53, 2, 21, 13] with applications to leakage-resilient secure multi-party computation [13].

We consider a notion of leakage-resilient secret sharing, where we allow the parties holding the shares to be entangled with each other. We now require the guarantee that security should still hold even if each share is individually leaked. Moreover, we consider a relaxed requirement where the shares are allowed to be quantum. Just like the works in the classical setting, we consider the bounded leakage model. That is, if the number of qubits of each share is m then we allow for some $\lfloor \frac{c}{n} \rfloor$ fraction of bits of leakage from each share, where c is some constant and n is the number of parties[4].

We show the following:

▶ **Theorem 4** (Informal). *There is a 2-out-n leakage-resilient secret sharing scheme with the following properties: (a) the shares are quantum, (b) the number of bits of leakage on each share is $\lfloor \frac{c \cdot m}{n} \rfloor$, where c is some constant and the size of each share is m qubits, and (c) the parties can share arbitrary amount of entanglement.*

[4] We set $m \gg n$.

In fact, our construction satisfies a stronger security guarantee where the adversary can receive $p(n)$ number of copies of its share, where $p(\cdot)$ is an arbitrary polynomial.

A recent interesting work by [20] also considers leakage-resilient secret sharing schemes with quantum shares. However, there are notable differences. Firstly, they consider the setting when there can be unbounded amount of classical[5] bits of leakage from each quantum share whereas we consider bounded leakage. On the other hand, we allow the parties to be entangled whereas they mainly focus on the LOCC setting. In fact, they show that it is not possible to achieve unbounded amount of leakage in the shared entanglement setting even with two parties; this is the reason we focus on the setting of bounded leakage. However, there seems to be a large gap between the amount of leakage leveraged in the impossibility result in [20][6] and the leakage that we tolerate in our feasibility result. It is an interesting problem to close the gap. Finally, we allow each party to get arbitrary polynomially many copies of its share whereas [20] doesn't satisfy this guarantee.

Paper Organization. We give a technical overview in Section 1.3. In Section 2 we formally define simultaneous state indistinguishability and give basic facts about it. In Section 3, we show SSI for Haar states. In Section 4 we show cryptographic applications of our result. For a full list of preliminaries and a list of missing proofs, please refer to the full version of the paper.

1.2 Subsequent Work

Studying the simultaneous Haar indistinguishabiliy problem could have further implications beyond unclonable cryptography. Indeed, inspired by our work, a recent subsequent work [4] studied a dual problem of simultaneous Haar indistinguishabiliy problem where two parties can communicate using classical channels but cannot share entanglement. They showed similar bounds and demonstrated applications to proving black-box separations in quantum cryptography. The combination of our results implies an interesting fact: entanglement and classical communication are resources that are individually insufficient to non-locally distinguish independent vs. identical Haar states, whereas together they suffice (indeed trivialize the problem) due to quantum teleportation. Exploring the relationship between the resources of entanglement and classical communication for different non-local tasks is an interesting research direction.

1.3 Technical Overview

1.3.1 Simultaneous Haar Indistinguishability

We formally define the notion of simultaneous state indistinguishability (SSI), of which simultaneous Haar indistinguishability is a special case. We consider a non-local distinguisher $(\text{Bob}, \text{Charlie}, \rho)$ where Bob and Charlie are spatially separated quantum parties who share an entangled state ρ. Given two distributions $\mathcal{D}_1, \mathcal{D}_2$ over bipartite states, we can write the distinguishing advantage of $(\text{Bob}, \text{Charlie}, \rho)$ as $d_{\mathsf{TV}}(x, x')$, where $x = (x_B, x_C)$ is the random variable corresponding to the output of Bob and Charlie when they get as input $\rho \otimes \psi$ where $|\psi\rangle$ is sampled from $\mathcal{D}_1$. Here, x_B refers to Bob's output and x_C refers to Charlie's output. Similarly, $x' = (x'_B, x'_C)$ is the random variable corresponding to Bob and Charlie's outputs when they receive $\rho \otimes \psi'$ where $|\psi'\rangle$ is sampled from $\mathcal{D}_2$.

[5] They also consider bounded quantum leakage in addition to the classical leakage.
[6] Their impossibility result seems to require $O(2^m)$ bits of total leakage from all the parties.

For fixed $\mathcal{D}_1, \mathcal{D}_2$ we can ask the question: *What is the maximal distinguishing advantage if Bob and Charlie are restricted to output n classical bits?*. We first limit our attention to a special case of this problem such that $n = 1$ as well as:

1. $\mathcal{D}_1$ outputs two identical Haar random states $|\psi\rangle \otimes |\psi\rangle$.
2. $\mathcal{D}_2$ outputs two independent Haar random states $|\psi_B\rangle \otimes |\psi_C\rangle$.

In both $\mathcal{D}_1$ and $\mathcal{D}_2$, the first half of the state will be given to Bob and the second half will be given to Charlie. Note that if we restrict our attention to Bob (or Charlie) alone, then the two cases are perfectly indistinguishable. Therefore, Bob and Charlie need to work collectively in order to achieve a non-trivial distinguishing advantage.

For now assume that the pre-shared entanglement consists of some arbitrary number (r) of EPR pairs, denoted by $|\mathsf{EPR}\rangle^{\otimes r}$. Let M and N be the measurements (formally POVM elements) applied by Bob and Charlie, respectively. It is without loss of generality to assume that M and N are projective. In this case, we can write the distinguishing advantage of Bob and Charlie as

$$\mathsf{Tr}\left[(M \otimes N)\left\{\mathsf{EPR}^{\otimes r} \otimes \left(\mathop{\mathbb{E}}_{|\psi\rangle} \psi \otimes \psi - \mathop{\mathbb{E}}_{\psi_B,\psi_C} \psi_B \otimes \psi_C\right)\right\}\right].$$

The second expectation equals the maximally mixed state $(\mathsf{id} \otimes \mathsf{id})/d^2$, where d is the dimension of the Haar random states. The first expectation equals the maximally mixed state over the symmetric subspace[7], which is $O(1/d)$ close to $(\mathsf{id} \otimes \mathsf{id} + F)/d^2$, where F is the operator that swaps two registers. Hence, we can approximate the advantage as

$$\frac{1}{d^2}\mathsf{Tr}\left[(M \otimes N)\left\{\mathsf{EPR}^{\otimes r} \otimes F\right\}\right].$$

We examine this expression in terms of tensor network diagrams[8] [49]. Up to normalization, the effect of the EPR pairs (followed by trace) is to connect the entangled registers of M and N in reverse (i.e. after partially transposing one of the projectors), whereas the effect of F (followed by trace) is to connect the registers of M and N containing the Haar states. Overall, we observe that the expression above equals

$$\frac{1}{2^r d^2}\mathsf{Tr}\left[M \cdot N^{\top_P}\right], \tag{1}$$

where $\top_P$ denotes the partial transpose operation. Notice that this is the Hilber-Schmidt inner-product of M and $N^{\top_P}$, hence by Cauchy-Schwarz we can bound it by

$$\frac{1}{2^r d^2}\left\|M\right\|_2 \cdot \left\|N^{\top_P}\right\|_2 = \frac{1}{2^r d^2}\left\|M\right\|_2 \cdot \left\|N\right\|_2 \leq \frac{1}{2^r d^2}\sqrt{2^r d} \cdot \sqrt{2^r d} = \frac{1}{d},$$

where we used the fact that $\|M\|_2^2$ equals the rank of M. Together with the previous approximation error we had, this gives us a bound of $O(1/d)$.

This bound is in fact tight, which can be seen by a simple attack where Bob and Charlie measure and output the first qubit of their input state. Moreover, this attack is unentangled, leading to the interesting conclusion that entangled attacks are no more powerful than unentangled attacks.

[7] See [37] for an introduction to the symmetric subspace.
[8] See [47] for an introduction to tensor network diagrams.

The Case of Many Copies. Now we generalize our argument to the case when Bob and Charlie each get t copies of their input. Similar to the $t = 1$ case, we can write the projection onto the symmetric subspace over t registers as

$$\Pi^t = \frac{1}{t!} \sum_{\sigma \in S_t} P_\sigma,$$

where P_σ is a register-wise permutation operator. Using this identity for Bob's and Charlie's registers alike, the independent case yields a sum of terms of the form $P_{\sigma_B} \otimes P_{\sigma_C}$. On the other hand, the identical case will give us a sum of P_σ for $\sigma \in S_{2t}$. We can match the coefficients up to $O(t^2/d)$ error, and thus we can approximate the advantage as

$$\frac{1}{d^{2t}} \mathsf{Tr} \left[(M \otimes N) \left\{ \mathsf{EPR}^{\otimes r} \otimes \sum_{\sigma \in S^*_{2t}} P_\sigma \right\} \right],$$

where S^*_{2t} is the set of permutations over $2t$ registers that cannot be expressed as a product of two permutations over t registers. A natural idea would be to bound this expression separately for each P_σ, but it would yield a factor of $\mathsf{poly}(t!)$ which blows up very fast. Our idea instead is to group permutations based on how far they are from a product permutation. In more detail, we define $S^{(s)}_{2t}$ as the set of permutations σ which obeys the identity

$$P_\sigma = (P_{\sigma_1} \otimes P_{\sigma_2}) \, P_{\sigma_s} \, (P_{\sigma_3} \otimes P_{\sigma_4})$$

for some permutations $\sigma_1, \sigma_2, \sigma_3, \sigma_4$ over t registers, where σ_s is a fixed permutation that swaps t of Bob's registers with t of Charlie's registers. Indeed, $S^{(s)}_{2t}$ is the set of permutations σ that make s *swaps* across Bob's and Charlie's registers. Moreover, by a combinatorial argument, we can compute the average of P_σ by averaging the identity above over $\sigma_1, \sigma_2, \sigma_3, \sigma_4$, so that

$$\sum_{\sigma \in S^{(s)}_{2t}} P_\sigma = C_s \left(\Pi^t_B \otimes \Pi^t_C \right) P_{\sigma_s} \left(\Pi^t_B \otimes \Pi^t_C \right),$$

where C_s is a constant that depends on s, t, and Π^t_B, Π^t_C are projections onto the symmetric subspace over Bob's and Charlie's t registers, respectively. Using the generalization of Equation (1) we show that

$$\frac{1}{d^{2t}} \mathsf{Tr} \left[(M \otimes N) \left\{ \mathsf{EPR}^{\otimes r} \otimes \sum_{\sigma \in S^{(s)}_{2t}} P_\sigma \right\} \right] \leq \binom{t}{s}^2 s! d^{-s},$$

and summing this over $s = 1, 2, \ldots, t$ gives us a bound of $O(t^2/d)$. Keep in mind that $s = 0$ is excluded from the sum because it corresponds to product permutations.

The Case of General Entanglement. Now suppose the entangled state $|\Omega\rangle$ shared between Bob and Charlie is arbitrary. Intuitively, we don't expect this relaxation to help the adversary a lot since the number of EPR pairs above was unbounded, yet it requires a proof to show this. Recall that $|\Omega\rangle$ can be written as

$$|\Omega\rangle = \sum_i \sqrt{\lambda_i} \, |u_i\rangle \, |v_i\rangle$$

for some choice of bases $(u_i), (v_i)$ for the registers of Bob and Charlie, known as the *Schmidt decomposition*. An equivalent way to write this is[9]

$$|\Omega\rangle = (\sigma^{1/2} \otimes V) \, |\mathsf{EPR}^{\otimes r}\rangle$$

for some density matrix σ, unitary V, and integer r. The unitary V can be safely ignored by changing the basis of Charlie's projection N. Using the cyclicity of trace we can absorb $\sigma^{1/2}$ in M and get a similar expression for the distinguishing advantage as the maximally entangled state, where M is replaced with $(\sigma^{1/2} \otimes \mathsf{id})M(\sigma^{1/2} \otimes \mathsf{id})/2^r$, with d' being the dimension of Bob's share of the entangled state. Following the same analysis as the EPR case results in a bound that scales with $\sqrt{d'}$. In order to get a bound that does not depend on the amount of entanglement we perform a more refined and involved analysis coupled with a more careful application of Cauchy-Schwarz, which yields a bound of $O(t^2/\sqrt{d})$ on the distinguishing advantage. An interesting open question is whether the gap between the EPR and non-EPR cases is inherent.

1.3.2 Applications

Unclonable Encryption. The search (weak) security of unclonable encryption (UE) was known since its formal introduction [17], yet strong (CPA-style/indistinguishability) security has been an open problem. There are fundamental reasons why this problem has been difficult, including the following:

1. Because the adversary learns the secret key in the challenge phase of the unclonable security experiment, it is hard to leverage traditional cryptographic tools – wherein revealing the secret key tantamounts to the compromise of security – in the construction of unclonable encryption.
2. There is a lack of straightforward equivalence between unpredictability and indistinguishability in the unclonability setting. The former is used to define CPA-style security and is not transitive, hence unfriendly to hybrid arguments.
3. Due to the simultaneous nature of the security experiment, extraction techniques that work for a single party often fail against two or more entangled parties.

To elaborate on the third bullet further, one can hope to deploy classical techniques for search-to-decision reductions in this setting. For instance, it has been shown that random oracles can be used to go from weak security to strong security in UE [6, 7]. In the plain model, a common classical tool is the Goldreich-Levin extraction technique [34], using which one can try the following folklore construction of UE:

1. The key consists of (k, r), where k is a key for a weakly secure UE scheme and $r \in \{0,1\}^n$ is a random string.
2. To encrypt a single-bit message $m \in \{0,1\}$, sample a random message $x \in \{0,1\}^n$, then output encryption of x using k, as well as $\langle r, x\rangle \oplus m$.
3. To decrypt, first use k to recover x using the decryption procedure of the weakly secure UE scheme and then recover m.

To prove unclonable security of this construction, one needs the *identical-challenge* version of simultaneous Goldreich-Levin, where Bob and Charlie will get the same r as challenge. This is unknown even though the independent-challenge version is known [44, 7].

[9] Here we are implicitly assuming that the dimensions of Bob and Charlie's registers both equal the same power of 2. This is merely for convenience and does not affect the analysis.

Our main insight is to make the string r in the key come in superposition, i.e. from a quantum state $\sum \alpha_r |r\rangle$. Intuitively, if Bob and Charlie were to measure r in the computational basis, then they would effectively receive independent values of r, meaning that we can hope to use independent-challenge Goldreich-Levin. Accordingly, we look for a state $\sum \alpha_r |r\rangle$ such that (1) Bob and Charlie cannot simultaneously distinguish whether or not this state has been measured in the computational basis, and (2) the computational basis measurement results in a uniform value of r.

Perhaps the most natural candidate for this task is to pick a Haar random state. This allows us to apply our simultaneous Haar indistinguishability result. Nonetheless, there still remain some technical challenges in the application of this concept. To begin with, we need to adapt the construction slightly to incorporate the newly acquired quantumness of r. Our solution is as follows:

1. The key is partially quantum:
   - The (classical) *encryption key* consists of $(k, x, \widetilde{b})$, where x is a random message and $\widetilde{b} \in \{0,1\}$ is a single-bit one-time-pad.
   - The (quantum) *decryption key* consists of k and a state $\sum \alpha_r |r\rangle |\langle r, x\rangle \oplus \widetilde{b}\rangle$, where $\sum \alpha_r |r\rangle$ is a Haar random state.
2. To encrypt a single-bit message $m \in \{0,1\}$, output encryption of x using k, as well as $\widetilde{b} \oplus m$.
3. To decrypt, first use k to recover x, then coherently recover $\widetilde{b}$ followed by m.

Using the simultaneous Haar indistinguishability, we can show that our construction is secure via the hybrid method. Note that we can do this because our notion of simultaneous-state-indistinguishability is strong enough that it is amenable to the use of hybrids.

In more detail, we reach an indistinguishable hybrid experiment where the Bob and Charlie get keys which use independently generated Haar random states. Equivalently, Bob gets $(r, \langle r, x\rangle \oplus \widetilde{b})$ and Charlie gets $(r', \langle r', x\rangle \oplus \widetilde{b})$ for independent r, r'.

Next, we move to a hybrid which is the weak security (i.e. search security) experiment of the underlying unclonable encryption scheme, so that Bob and Charlie need to output x each given the key k. Unfortunately, even though r and r' are independent in the previous hybrid, independent-challenge simultaneous Goldreich-Levin [44, 7] is insufficient due to the correlation between the bits $b = \langle r, x\rangle \oplus \widetilde{b}$ and $b' = \langle r', x\rangle \oplus \widetilde{b}$, namely $b \oplus b' = \langle r, x\rangle \oplus \langle r', x\rangle$. To overcome this issue, we prove exactly what we need, which we call *correlated simultaneous quantum Goldreich-Levin*[10] [11] (Lemma 30), which can be summarized as follows:

> Correlated Goldreich-Levin: Suppose that Bob is given input (r, b) and Charlie is given input (r', b'), where r, r' are independent strings and b, b' are uniform bits satisfying the correlation $b \oplus b' = \langle r, x\rangle \oplus \langle r', x\rangle$. If (Bob, Charlie) can output $(\langle r, x\rangle, \langle r', x\rangle)$ with probability $1/2 + 1/\mathsf{poly}$, then there is an extractor $(\mathrm{ExtBob}, \mathrm{ExtCharlie})$ that extracts (x, x) from (Bob, Charlie) with probability $1/\mathsf{poly}$.

To prove this lemma we first tackle the correlation between b and b'. Consider $\widetilde{\text{Bob}}$ who takes as input r, samples b himself uniformly, and runs Bob on input (r, b) to obtain b_B; similarly consider $\widetilde{\text{Charlie}}$ who takes r' as input, samples b' and runs Charlie on input (r', b') to obtain b_C. Now that the input bits b, b' are uncorrelated, $(\widetilde{\text{Bob}}, \widetilde{\text{Charlie}})$, who output (b_B, b_C), are expected to have worse success probability than (Bob, Charlie). However, we can in turn relax the success criterion for $(\widetilde{\text{Bob}}, \widetilde{\text{Charlie}})$ to merely output bits (b_B, b_C) that

[10] As a side note, this lemma resolves an open question in [44], implying that their construction achieves a more desirable notion of security.

[11] Previously, the work of [3] explicitly stated the correlated Goldreich-Levin problem over large finite fields as a conjecture.

satisfy $b_B \oplus b_C = \langle r, x\rangle \oplus \langle r', x\rangle$ in order to counteract this lack of correlation. In other words, now $(\widetilde{\text{Bob}}, \widetilde{\text{Charlie}})$ are additionally allowed to be *both incorrect*. Indeed, we show that the success probability of $(\widetilde{\text{Bob}}, \widetilde{\text{Charlie}})$ in this case is at least that of (Bob, Charlie), i.e. $1/2 + 1/\mathsf{poly}$. To show this fact, we define E as the event that $b \oplus b' = \langle r, x\rangle \oplus \langle r', x\rangle$. Conditioned on E, Bob and Charlie will output $(\langle r, x\rangle, \langle r', x\rangle)$ with probability $1/2 + 1/\mathsf{poly}$ by our assumption. In addition, the event E is independent of Bob's (or Charlie's) marginal output due to the fact that the players' correlation satisfies no-signalling. To see this, notice that the bits b and b' can each independently control the event E. We utilize this important observation to show the desired result. Another way to interpret this reduction is as follows: the correlation that (Bob, Charlie) require in order to output $(\langle r, x\rangle, \langle r', x\rangle)$ appears as a correlation in the output of $(\widetilde{\text{Bob}}, \widetilde{\text{Charlie}})$, who take uncorrelated bits as input.

After this reduction, it seems that we still cannot use the original independent-challenge simultaneous Goldreich-Levin because of the relaxed success criterion above. Luckily, by examining the proof of [7] we see that this condition is sufficient without additional work for the existence of (ExtBob, ExtCharlie) who can extract (x, x) simultaneously.

Many-Copy Security. For t-copy security, where Bob and Charlie get t copies of the decryption key in the unclonable security experiment, we need t to be at most linear in n, for otherwise Bob and Charlie can learn x using Gaussian elimination. In the proof, we similarly reach a hybrid where the Haar random states given to Bob and Charlie are independent. Then, we need an extra step where we switch to a hybrid in which Bob gets as input $(r_i, \langle r_i, x\rangle \oplus \widetilde{b})$ for independent samples $r_1, \ldots, r_t$ (instead of $(r_1, \ldots, r_t)$ being generated from t copies of a Haar random state) and similarly Charlie gets $(r'_i, \langle r'_i, x\rangle \oplus \widetilde{b})$ for independently generated $r'_1, \ldots, r'_t$. We show that the success probability of Bob and Charlie does not decrease from this change. To show this, we argue that that given $(r_i, \langle r_i, x\rangle \oplus \widetilde{b})$ Bob can prepare

$$\left(\sum_r \alpha_r \left|r\right\rangle \left|\langle r, x\rangle \oplus \widetilde{b}\right\rangle\right)^{\otimes t}$$

where $|\varphi\rangle = \sum_r \alpha_r |r\rangle$ is a Haar random state. In the expression above, $|r\rangle$ corresponds to the register that holds Goldreich-Levin samples, which are generated from t copies of a Haar random state rather than t independent samples. Recall that $\varphi^{\otimes t}$ can be written as a random vector in the *type basis*. Using this fact, Bob can coherently apply a random permutation $\sigma \in S_t$ to the values $|r_i, \langle r_i, x\rangle \oplus \widetilde{b}\rangle$, after which he can uncompute the permutation σ. We can argue similarly for Charlie, since their inputs originate from independent Haar distributions. This argument in fact requires the strings r_i to be distinct, which fortunately holds with high probability by the birthday bound.

Note that the input above given to Bob can be thought of as $(r, \langle r, x\rangle \oplus \widetilde{b})$ alongside $t - 1$ random samples of $(r_i, \langle r_i, x\rangle)$. In the final step, we apply our correlated Goldreich-Levin result in the presence of this extra information to reach the search security experiment for the weakly secure UE scheme. In this experiment, the extra information can be guessed by Bob and Charlie, hence if t is bounded by a linear function of n the security still holds.

Single-Decryptor Encryption. Single-decryptor encryption is a primitive that closely resembles unclonable encryption. It is an encryption scheme in which the decryption key is unclonable. In the security experiment, Alice tries to clone a quantum decryption key and split it between Bob and Charlie, who then try to decrypt a ciphertext they receive using their shares of the key. Depending on the correlation of these ciphertexts, one can

define *identical-challenge* or *independent-challenge* security. We adapt our construction of unclonable encryption with quantum decryption keys to construct single-decryptor encryption with quantum ciphertexts. Our construction can be summarized as follows:

1. The encryption key consists of a key k as well as a random message x for a weakly secure UE scheme. The quantum decryption key contains k and encryption of x using k.
2. To encrypt a one-bit message m, output k as well as $\sum \alpha_r |r\rangle |\langle r, x\rangle \oplus m\rangle$, where $\sum \alpha_r |r\rangle$ is a Haar random state.
3. To decrypt, first recover x and then coherently compute m.

We can show that this construction is secure if Bob and Charlie are given t copies of the same ciphertext for $t = O(|x|)$. The proof is nearly identical to the security proof of our UE construction above. For more clear exposition, in the technical sections we first present our construction of single-decryptor encryption (Section 4.1), followed by that of unclonable encryption (Section 4.2).

Classical-Leakage-Resilient Secret-Sharing. As another application of simultaneous Haar indistinguishability, we construct a 2-out-of-n quantum secret-sharing scheme for a single-bit classical message. The construction is as follows:

1. The shares of bit $b = 0$ are identical Haar random states $|\psi\rangle$ and the shares of $b = 1$ are independent Haar random states $|\psi_i\rangle$. In addition, we give t copies of their share to all parties $1, 2, \ldots, n$.
2. Any two parties i, j can recover the message by applying t SWAP tests between their secrets. If $m = 0$, then all the tests will succeed. On the other hand, if $m = 1$, then with high probability the independent Haar random states held by i and j will be almost orthogonal, therefore the number of SWAP tests that pass will be concentrated near $t/2$ by a Chernoff bound.

Formally, we show that m remains hidden in the presence of ℓ-bits of classical leakage from each party. This amounts to showing a many-party and many-bit generalization of simultaneous Haar indistinguishability (SHI), that is, either all parties get (copies of) the same Haar random state or they get (copies of) independent Haar random states. Firstly, we go from 1-bit SHI to ℓ-bit SHI for 2 parties. This can be achieved by a union bound, which incurs a multiplicative loss of $2^{2\ell}$ in security. And then we show an equivalence between SHI in the cases of (1) 2 parties each getting $O(nt)$ copies and (2) $O(n)$ parties each getting t copies. This can be seen as distributing a fixed number of Haar random states among more parties. In the proof we use $O(\log n)$ hybrids, doubling the number of parties at each hybrid. We show by an additional hybrid argument that we incur a multiplicative loss of $O(1)$ at each step, hence a multiplicative loss of $O(n)$ in total after $O(\log n)$ steps. Putting everything together, we show that we can allow $\ell = O(\log d/n)$ bits of leakage from each party, where $\log d$ is the number of qubits of each copy of a share. The number of copies (t) given to each party can be an arbitrary polynomial, which lets us amplify the correctness of the scheme. An interesting open question is to get rid of the exponential dependence on the number of bits leaked (ℓn), which would imply that our construction tolerates unbounded polynomial leakage.

1.4 Notation

We write $\log := \log_e$ to denote the natural logarithm. We write $[n] := \{1, \ldots, n\}$. We use the notation $\langle \cdot, \cdot \rangle : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ to denote the inner product over $\mathbb{F}_2^n$, i.e. for classical strings $x, y \in \{0,1\}^n$ we have $\langle x, y\rangle := \sum_{i=1}^n x_i y_i \pmod 2$. We denote the set of d-dimensional pure quantum states by $\mathcal{S}(\mathbb{C}^d) := \left\{|\psi\rangle \in \mathbb{C}^d \ : \ \langle\psi|\psi\rangle = 1\right\}$. We sometimes write $\psi := |\psi\rangle\langle\psi|$ for simplicity. For a complex matrix (or operator) A, we write $A^\top$ to

denote its transpose and $\overline{A}$ to denote its entry-wise complex conjugation, both with respect to the computational basis. $d_{\mathsf{TV}}(x, x')$ denotes the total-variation distance between random variables x, x'.

2 Simultaneous State Indistinguishability

Terminology. Below, $\mathcal{D}$ represents a probability distribution over pure states. Particularly, we will consider bipartite pure states $|\psi\rangle_{BC}$ and non-local adversaries $\mathcal{A} = (\mathcal{B}, \mathcal{C}, \rho_{BC})$ as distinguishers, where the B register will be given to $\mathcal{B}$ and the C register will be given to $\mathcal{C}$.

2.1 Definitions

▶ Remark 5. A distribution $\mathcal{D}$ has a unique representation as a quantum mixed state $\rho_{\mathcal{D}}$, whereas a quantum mixed state can represent many distributions. Accordingly, we can apply our results to mixed states ρ for which we can find an appropriate distribution $\mathcal{D}'$ that satisfies $\rho_{\mathcal{D}'} = \rho$.

▶ **Definition 6** (Simultaneous State Indistinguishability)**.** *We say that two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ are* ε-simultaneous state indistinguishable *(ε-SSI) against a non-local adversary $\mathcal{A} = (\mathcal{B}, \mathcal{C}, \rho_{BC})$ if the following holds for every pair of bits b_1, b_2:*

$$\left| \mathsf{Pr}\left[(b_1, b_2) \leftarrow \mathcal{A}(\psi_{BC}) \ : \ |\psi\rangle_{BC} \leftarrow \mathcal{D}_1\right] - \mathsf{Pr}\left[(b_1, b_2) \leftarrow \mathcal{A}(\psi_{BC}) \ : \ |\psi\rangle_{BC} \leftarrow \mathcal{D}_2\right] \right| \leq \varepsilon$$

Here, $\mathcal{B}$ gets the register B of ψ and $\mathcal{C}$ gets the register C.

Of interest to us is the case when $\mathcal{D}_1 = \mathcal{D}^{\mathsf{id}}$ and $\mathcal{D}_2 = \mathcal{D}^{\mathsf{ind}}$, where we define $\mathcal{D}^{\mathsf{id}}$ and $\mathcal{D}^{\mathsf{ind}}$ as follows[12]:

$(\mathcal{D}^{\mathsf{id}})$: Sample $|\psi\rangle \leftarrow \mathcal{D}$ and output $(|\psi\rangle_B \otimes |\psi\rangle_C)$

$(\mathcal{D}^{\mathsf{ind}})$: Sample $|\psi\rangle, |\psi'\rangle \leftarrow \mathcal{D}$ and output $(|\psi\rangle_B \otimes |\psi'\rangle_C)$, i.e. $\mathcal{D}^{\mathsf{ind}} = \mathcal{D} \times \mathcal{D}$.

In this case, we refer to the above notion as $(\varepsilon, \mathcal{D})$-simultaneous state indistinguishability ($(\varepsilon, \mathcal{D})$-SSI). Note that up to a constant factor this is equivalent to the output distributions of $\mathcal{A}$ with respect to $\mathcal{D}_1, \mathcal{D}_2$ having total variation distance ε. Also note that we can fix the bits b_1, b_2 without loss of generality.

▶ **Definition 7** ($(\varepsilon, \mathcal{D})$-SSI)**.** *We say that* $(\varepsilon, \mathcal{D})$-SSI *holds if $\mathcal{D}^{\mathsf{id}}$ and $\mathcal{D}^{\mathsf{ind}}$ defined above are ε-SSI against all non-local adversaries $(\mathcal{B}, \mathcal{C}, \rho)$.*

▶ Remark 8. When it is clear from the context, we omit the mention of the registers B and C.

2.1.1 SSI as a metric

This notion defines a metric over bipartite states, namely $T_{SSI}(\rho_0, \rho_1)$ is the smallest value of ε such that ρ_0 and ρ_1 are ε simultaneously indistinguishable. Equivalently,

$$T_{SSI}(\rho_0, \rho_1) := \sup_{(\mathcal{B}, \mathcal{C}, \rho)} \left| \Pr\left[(1,1) \leftarrow (\mathcal{B} \otimes \mathcal{C})(\rho \otimes \rho_0)\right] - \Pr\left[(1,1) \leftarrow (\mathcal{B} \otimes \mathcal{C})(\rho \otimes \rho_1)\right] \right|. \quad (2)$$

It is easy to see that this is a valid metric[13].

Since any binary measurement is a linear combination of projective measurements by the Spectral Theorem, we have the following fact.

[12] Here id stands for *identical* and ind stands for *independent*.

[13] Technically, a pseudometric since two different states may have distance 0.

▶ **Lemma 9.** *$T_{SSI}(\cdot,\cdot)$ can be equivalently defined by restricting $\mathcal{B},\mathcal{C}$ to be projective measurements.*

2.1.2 Extending to Many-Bit Output

We can extend the definition of SSI such that the non-local adversary can output many bits in each register. We then require that the total variation distance between the collective outputs of $\mathcal{B}$ and $\mathcal{C}$ for the two distributions is small.

▶ **Definition 10** (($\varepsilon,\mathcal{D},n$)-SSI)**.** *We say that $(\varepsilon,\mathcal{D},n)$-SSI holds if for any nonlocal adversary $\mathcal{A}=(\mathcal{B},\mathcal{C},\rho)$ and any $S\subseteq\{0,1\}^n\times\{0,1\}^n$ we have*

$$\left|\Pr\left[(x_1,x_2)\in S \;:\; \substack{|\psi\rangle\leftarrow\mathcal{D}\\(x_1,x_2)\leftarrow\mathcal{A}(\psi\otimes\psi)}\right]-\Pr\left[(x_1,x_2)\in S \;:\; \substack{|\psi\rangle,|\psi'\rangle\leftarrow\mathcal{D}\\(x_1,x_2)\leftarrow\mathcal{A}(\psi\otimes\psi')}\right]\right|\leq\varepsilon$$

Note that $(\varepsilon,\mathcal{D},2)$-SSI and $(\varepsilon,\mathcal{D})$-SSI are equivalent up to a constant factor on ε. There is a straightforward relation between $(\varepsilon,\mathcal{D})$-SSI and $(\varepsilon,\mathcal{D},n)$-SSI using the union bound, which we formally state below.

▶ **Lemma 11.** *Suppose $(\varepsilon,\mathcal{D})$-SSI holds, then $(2^{2n-1}\varepsilon,\mathcal{D},n)$-SSI holds for all $n\in\mathbb{N}$.*

Proof. For any non-local adversary $\mathcal{A}=(\mathcal{B},\mathcal{C},\rho)$ and any $y,y'\in\{0,1\}^n$, we have

$$\left|\Pr\left[(y,y')\leftarrow\mathcal{A}(\psi\otimes\psi)\;:\;|\psi\rangle\leftarrow\mathcal{D}\right]-\Pr\left[(y,y')\leftarrow\mathcal{A}(\psi\otimes\psi')\;:\;|\psi\rangle,|\psi'\rangle\leftarrow\mathcal{D}\right]\right|\leq\varepsilon$$

by $(\varepsilon,\mathcal{D})$-SSI. This is because $\mathcal{B}$ ($\mathcal{C}$) can associate y (respectively, y') with 0 and every other string with 1. By summing over all y,y' and dividing by 2 we get the desired result. ◀

2.1.3 Extending to Many Copies

For a distribution $\mathcal{D}$, denote by $\mathcal{D}^t$ the distribution that samples $|\psi\rangle\leftarrow\mathcal{D}$ and outputs $|\psi\rangle^{\otimes t}$. Then, we can consider $(\varepsilon,\mathcal{D}^t)$-SSI or $(\varepsilon,\mathcal{D}^t,n)$-SSI as extensions where $\mathcal{B}$ and $\mathcal{C}$ each get t copies of their respective inputs.

2.1.4 Extending to Many Parties

We can consider as the distinguisher a q-party non-local adversary $\mathcal{A}=(\mathcal{A}_1,\ldots,\mathcal{A}_q,\rho_{A_1\cdots A_q})$. By giving every party t copies of a quantum state generated either identically or independently we can generalize Definition 10:

▶ **Definition 12** (($\varepsilon,\mathcal{D},n$)-SSI against q Parties)**.** *We say that $(\varepsilon,\mathcal{D},n)$-SSI holds* against q parties *if for any q-party nonlocal adversary $\mathcal{A}=(\mathcal{A}_1,\ldots,\mathcal{A}_q,\rho_{A_1\cdots A_q})$ and any $S\subseteq\{0,1\}^{nq}$ we have*

$$\left|\Pr\left[(x_1,\ldots,x_q)\in S\;:\;\substack{|\psi\rangle\leftarrow\mathcal{D}\\(x_1,\ldots,x_q)\leftarrow\mathcal{A}(\psi_{A_1}\otimes\cdots\otimes\psi_{A_q})}\right]\right.$$
$$\left.-\Pr\left[(x_1,\ldots,x_q)\in S\;:\;\substack{|\psi_1\rangle,\ldots,|\psi_q\rangle\leftarrow\mathcal{D}\\(x_1,\ldots,x_q)\leftarrow\mathcal{A}((\psi_1)_{A_1}\otimes\ldots\otimes(\psi_q)_{A_q})}\right]\right|\leq\varepsilon$$

In general, SSI against many parties is weaker than regular SSI with the same number of copies and the same total output length, which we state formally below.

▶ **Lemma 13.** *Suppose that $(\varepsilon,\mathcal{D}^{qt},qn)$-SSI holds (against 2 parties), then $(2q\varepsilon,\mathcal{D}^t,n)$-SSI holds against q parties.*

Combining Lemmas 11 and 13 yields the following corollary:

▶ **Corollary 14.** *Suppose that $(\varepsilon,\mathcal{D}^{qt})$-SSI holds, then $(2^{2qn}q\varepsilon,\mathcal{D}^t,n)$-SSI holds against q parties.*

2.2 Distinguishing Bell-States

In this section, we use the Bell basis over a 2-qubit system as a warm-up example to demonstrate some facts about simultaneous state indistinguishability. By the Bell basis we mean

$$\left\{ \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right), \frac{1}{\sqrt{2}}\left(|00\rangle - |11\rangle\right), \frac{1}{\sqrt{2}}\left(|{+}{+}\rangle + |{-}{-}\rangle\right), \frac{1}{\sqrt{2}}\left(|{+}{+}\rangle - |{-}{-}\rangle\right) \right\}.$$

Note that the Bell basis is symmetric for the purposes of this section, meaning any fact we show about a pair of Bell states will also apply to any other pair of Bell states. We start off by demonstrating that the non-local distance $T_{SSI}(\cdot,\cdot)$ can be strictly less than LOCC distance. Recall that two orthogonal Bell states can be perfectly distinguished using LOCC measurements.

▶ **Lemma 15** (Comparison to LOCC). *Let $|\Phi^+\rangle, |\Phi^-\rangle$ be orthogonal Bell states. Then, $T_{SSI}(|\Phi^+\rangle\langle\Phi^+|, |\Phi^-\rangle\langle\Phi^-|) = 1/2$.*

Proof. It suffices to consider projective $\mathcal{B}, \mathcal{C}$. After that, the only non-trivial case is when they both are rank-1 projections, in which case it is easy to show

$$\mathsf{Tr}\left((\mathcal{B}\otimes\mathcal{C})\left(|\Phi^+\rangle\langle\Phi^+| - |\Phi^-\rangle\langle\Phi^-|\right)\right) \leq \mathsf{Tr}\left((\mathcal{B}\otimes\mathcal{C})|\Phi^+\rangle\langle\Phi^+|\right) \leq 1/2. \qquad \blacktriangleleft$$

Next, we show that a non-local distinguisher with an arbitrary number of EPR pairs is no more powerful than unentangled distinguishers for the same problem.

▶ **Lemma 16** (Entanglement has no effect). *Let $|\Phi^+\rangle, |\Phi^-\rangle$ be orthogonal Bell states and let $|\mathsf{EPR}_n\rangle$ denote n EPR pairs. Then, $T_{SSI}(|\Phi^+\rangle\langle\Phi^+| \otimes |\mathsf{EPR}_n\rangle\langle\mathsf{EPR}_n|, |\Phi^-\rangle\langle\Phi^-| \otimes |\mathsf{EPR}_n\rangle\langle\mathsf{EPR}_n|) = 1/2$.*

2.3 Impossibility Results about SSI

There are many distributions that are simultaneously distinguishable. We give some examples below by identifying $\mathcal{D}$ for which $(\varepsilon, \mathcal{D})$-SSI does not hold for small ε.

▷ Claim 17. Suppose $\mathcal{D}$ is a (classical) distribution on $\{0,1\}^n$ such that $\mathsf{Pr}[y_1 \neq y_2 : y_1, y_2 \leftarrow \mathcal{D}] \geq c$, for some $c = c(n)$. Then, $(\varepsilon, \mathcal{D})$-simultaneous state indistinguishability does not hold for any $\varepsilon < c/2n$.

▷ Claim 18. Let $m, n \in \mathbb{N}$ and $m \geq n$. Let $C \in \mathcal{L}(\mathbb{C}^{2^n}, \mathbb{C}^{2^m})$ be a Clifford circuit. That is, C appends $m-n$ qubits initialized to zeros to its input and applies a sequence of Clifford gates. Define $\mathcal{D}_1$ as follows: sample $r \xleftarrow{\$} \{0,1\}^{n-1}$ and then output $|\psi_r\rangle_{BC} = C|r||0\rangle$. Define $\mathcal{D}_2$ as follows: sample $r \xleftarrow{\$} \{0,1\}^{n-1}$ and then output $|\psi_r\rangle_{BC} = C|r||1\rangle$. Then, the distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ are ε-simultaneous state distinguishable for all $\varepsilon < \frac{1}{2}$.

3 Simultaneous Haar Indistinguishability

We show that SSI can be instantiated using Haar random states.

3.1 Single-Copy Case

▶ **Theorem 19** (Simultaneous Haar Indistinguishability (SHI)). *$(\varepsilon, \mathcal{H}_{2^n})$-simultaneous state indistinguishability holds for $\varepsilon = O(1/2^{n/2})$.*

▶ Remark 20. The entangled state held by $\mathcal{A}$ above can have arbitrary dimension. In particular, the proof shows that an arbitrary number of EPR pairs has no asymptotic advantage in distinguishing identical vs. independent Haar states, for the advantage is $O(1/d)$ in both cases.

3.2 Generalization to Many Copies

We generalize Theorem 19 to show indistinguishability when the non-local distinguisher is given many copies of the input state. The proof has the same overall structure, but requires some additional ideas. We start by listing a useful lemma.

▶ **Lemma 21.** *Let $S_{2t}^{(s)} \subset S_{2t}$ be the set of permutations σ over $[2t]$ such that*

$$|\{i \in [t] \ : \ \sigma(i) \notin [t]\}| = s.$$

Define P_σ as the permutation operator over registers $X_1 \dots X_t Y_1 \dots Y_t$, where we match $[t]$ with $X_1 \dots X_t$ and $[2t] \setminus [t]$ with $Y_1 \dots Y_t$. Then,

$$\sum_{\sigma \in S_{2t}^{(s)}} P_\sigma = (t!)^2 \binom{t}{s}^2 \left(\Pi_{\mathsf{Sym}}(X_1 \dots X_t) \otimes \Pi_{\mathsf{Sym}}(Y_1 \dots Y_t)\right) P_{\sigma_s} \left(\Pi_{\mathsf{Sym}}(X_1 \dots X_t) \otimes \Pi_{\mathsf{Sym}}(Y_1 \dots Y_t)\right),$$

where $\sigma_s \in S_{2t}^{(s)}$ is the permutation that swaps i with $t+i$ for $i \in [s]$.

▶ **Theorem 22** (t-Copy Simultaneous Haar Indistinguishability (SHI))**.** *Let $\varepsilon = o(1)$ and $t = \varepsilon 2^{n/4}$. Let $\mathcal{H}_{2^n}^t$ be the distribution defined by sampling t copies of a state from $\mathcal{H}_{2^n}$. Then, $(O(\varepsilon^2), \mathcal{H}_{2^n}^t)$-simultaneous state indistinguishability holds.*

4 Applications

We present applications of simultaneous Haar indistinguishability (Section 3) to single-decryptor encryption (Section 4.1) and unclonable encryption (Section 4.2). Ordinarily, single-decryptor encryption is defined with classical ciphertexts and quantum decryption keys, whereas unclonable encryption is defined using quantum ciphertexts and classical encryption/decryption keys. We achieve relaxed notions of both primitives above: namely, we additionally allow quantum ciphertexts in single-decryptor encryption and quantum decryption keys in unclonable encryption.

In Section 4.3, we show how to construct leakage-resilient quantum secret sharing of classical messages, which additionally guarantees security against an eavesdropper that learns classical leakage of all the shares.

4.1 Single-Decryptor Encryption with Quantum Cipertexts

4.1.1 Definitions

We adopt the definition of single-decryptor encryption by [33] to the setting where the ciphertexts can be quantum.

▶ **Definition 23** (Single-Decryptor Encryption)**.** *A* single-decryptor encryption *(SDE) scheme is a tuple of QPT algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*:*

- $\mathsf{Gen}(1^\lambda)$ *takes as input a security paramter. It outputs a classical encryption key* ek *and a one-time quantum decryption key* $|🔑\rangle$.
- $\mathsf{Enc}(\mathsf{ek}, m)$ *takes as input an encryption key* ek*, and a classical message m. It outputs a quantum ciphertext* $|\mathsf{CT}\rangle$*. We require that* Enc *is pseudo-deterministic.*
- $\mathsf{Dec}(|🔑\rangle, |\mathsf{CT}\rangle)$ *takes as input a quantum decryption key* $|🔑\rangle$*, a quantum ciphertext* $|\mathsf{CT}\rangle$ *and outputs a classical message m.*

▶ **Definition 24** (Correctness). *A SDE scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with quantum ciphertexts is* correct *if for any security parameter* λ *and any message* m *we have*

$$\Pr\left[m' = m \ : \ \begin{matrix}(\mathsf{ek}, |\mathsf{sk}\rangle) \leftarrow \mathsf{Gen}(1^\lambda) \\ |\mathsf{CT}\rangle \leftarrow \mathsf{Enc}(\mathsf{ek}, m) \\ m' \leftarrow \mathsf{Dec}(|\mathsf{sk}\rangle, |\mathsf{CT}\rangle)\end{matrix}\right] \geq 1 - \mathsf{negl}(\lambda).$$

Before defining security, we introduce a notation $\mathsf{Enc}^T(\mathsf{ek}, m)$, which means sampling randomness r and running $\mathsf{Enc}(\mathsf{ek}, m; r)$ for T times.

▶ **Definition 25** (Security of SDE). *An SDE scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is called* (information-theoretically) secure against identical ciphertexts *if for any cloning adversary* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ *and any pair of messages* (m_0, m_1) *we have*

$$\Pr\left[b_{\mathcal{B}} = b_{\mathcal{C}} = b \ : \ \begin{matrix}(\mathsf{ek}, |\mathsf{sk}\rangle) \leftarrow \mathsf{Gen}(1^\lambda) \\ \rho_{BC} \leftarrow \mathcal{A}(|\mathsf{sk}\rangle) \\ b \xleftarrow{\$} \{0,1\}, \quad |\mathsf{CT}\rangle^{\otimes 2} \leftarrow \mathsf{Enc}^2(\mathsf{ek}, m_b) \\ b_{\mathcal{B}} \leftarrow \mathcal{B}(|\mathsf{CT}\rangle, \rho_B), \quad b_{\mathcal{C}} \leftarrow \mathcal{C}(|\mathsf{CT}\rangle, \rho_C)\end{matrix}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda),$$

where ρ_E *denotes the* E *register of the bipartite state* ρ_{BC} *for* $E \in \{B, C\}$.

▶ Remark 26 (Identical Ciphertexts). In Definition 25 we consider security against identical ciphertexts. One can similarly define security against ciphertexts that are independently generated. This alternate definition was achieved in the plain model by [7], and it only requires independent-challenge Goldreich-Levin.

▶ Remark 27. Note that Definition 25 need not be physical for an arbitrary scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ without the requirement that Enc is pseudo-deterministic due to the fact that $|\mathsf{CT}\rangle$ may be unclonable even for the encryptor. Nonetheless, our construction satisfies this condition, with the classical randomness of Enc being used for sampling a Haar random state.

Below, we consider a stronger security definition where many copies of the quantum ciphertext are given to the adversary. This matches the case of classical ciphertext more closely, since classical ciphertexts can be cloned arbitrarily. Another implication of this stronger definition is that security holds against adversaries who can clone the quantum ciphertexts.

▶ **Definition 28** (t-Copy Security of SDE). *An SDE scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is called* (information-theoretically) t-copy secure against identical ciphertexts *if for any cloning adversary* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ *and any pair of messages* (m_0, m_1) *we have*

$$\Pr\left[b_{\mathcal{B}} = b_{\mathcal{C}} = b \ : \ \begin{matrix}(\mathsf{ek}, |\mathsf{sk}\rangle) \leftarrow \mathsf{Gen}(1^\lambda) \\ \rho_{BC} \leftarrow \mathcal{A}(|\mathsf{sk}\rangle) \\ b \xleftarrow{\$} \{0,1\}, \quad |\mathsf{CT}\rangle^{\otimes 2t} \leftarrow \mathsf{Enc}^{2t}(\mathsf{ek}, m_b) \\ b_{\mathcal{B}} \leftarrow \mathcal{B}(|\mathsf{CT}\rangle^{\otimes t}, \rho_B), \quad b_{\mathcal{C}} \leftarrow \mathcal{C}(|\mathsf{CT}\rangle^{\otimes t}, \rho_C)\end{matrix}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda),$$

where ρ_E *denotes the* E *register of the bipartite state* ρ_{BC} *for* $E \in \{B, C\}$.

4.1.2 Construction

Let $(\mathsf{Gen}_{\mathsf{UE}}, \mathsf{Enc}_{\mathsf{UE}}, \mathsf{Dec}_{\mathsf{UE}})$ be a one-time unclonable encryption scheme with C^n-weak unclonable security and message space $\mathcal{M} \subseteq \{0,1\}^n$, with $n = \mathsf{poly}(\lambda)$, where $C \in (1/2, 1)$ is a constant. Let $\mathcal{D} = \mathcal{H}_{2^n}$, so that:

1. $(\varepsilon, \mathcal{D}^t)$-SSI holds for some $\varepsilon = \mathsf{negl}(\lambda)$ by Theorem 22 as long as $t^2/2^{n/2}$ is negligible.
2. $\mathbb{E}_{|\psi\rangle \leftarrow \mathcal{D}} |\psi\rangle\langle\psi| = \mathsf{id}_{2^n}$.

We construct SDE for single-bit messages ($m \in \{0,1\}$) secure against identical ciphertexts as follows:

- $\mathsf{Gen}(1^\lambda)$ samples a random message $x \xleftarrow{\$} \mathcal{M}$ and a key $k \leftarrow \mathsf{Gen}_{\mathsf{UE}}(1^\lambda)$. It computes $|\psi\rangle \leftarrow \mathsf{Enc}_{\mathsf{UE}}(k, x)$. It outputs an encryption key $\mathsf{ek} = (k, x)$ and a decryption key $|🔑\rangle = |\psi\rangle$.
- $\mathsf{Enc}(\mathsf{ek}, m)$ samples $|\varphi\rangle = \sum_y \alpha_y |y\rangle \leftarrow \mathcal{D}$. It parses $\mathsf{ek} = (k, x)$ and computes the state $|\phi\rangle = \sum_y \alpha_y |y\rangle |\langle y, x\rangle \oplus m\rangle$. It outputs a quantum ciphertext $|\mathsf{CT}\rangle = (k, |\phi\rangle)$. Note that Enc is pseudo-deterministic given that it can sample from $\mathcal{D}$ using classical randomness.
- $\mathsf{Dec}(|🔑\rangle, |\mathsf{CT}\rangle)$ parses $|\mathsf{CT}\rangle = (k, |\phi\rangle)$. It computes $x \leftarrow \mathsf{Dec}_{\mathsf{UE}}(k, |\phi\rangle)$. It computes $U_x |\psi\rangle$ and measures the second register to obtain m, where U_x is the unitary defined as $U_x |y\rangle |z\rangle = |y\rangle |z \oplus \langle y, x\rangle\rangle$. It outputs m.

▶ Remark 29. Since t is bounded (see Theorem 34), we can use a $2t$-state design to instantiate the Haar random state used in the construction. We can similarly use a $2t$-state design to instantiate our construction of unclonable encryption in Section 4.2.

From the correctness of $(\mathsf{Gen}_{\mathsf{UE}}, \mathsf{Enc}_{\mathsf{UE}}, \mathsf{Dec}_{\mathsf{UE}})$, it follows that Dec recovers m (with probability negligibly close to 1) from $|\mathsf{CT}\rangle$, where $|\mathsf{CT}\rangle$ is an encryption of m.

4.1.3 Security Proof

We first show a lemma that is needed in our security proof:

▶ **Lemma 30** (Simultaneous Quantum Goldreich-Levin with Correlated Input)**.** *Let $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ be a cloning adversary that given*[14] *$(\rho, (\sigma, (y_\mathcal{B}, b_\mathcal{B}), (y_\mathcal{C}, b_\mathcal{C})))$, where $y_\mathcal{B}, y_\mathcal{C} \in \{0,1\}^n$ are i.i.d. uniform strings and $b_\mathcal{B}, b_\mathcal{C} \in \{0,1\}$ are random bits satisfying $b_\mathcal{B} \oplus b_\mathcal{C} = \langle y_\mathcal{B}, x\rangle \oplus \langle y_\mathcal{C}, x\rangle$, can output $(\langle y_\mathcal{B}, x\rangle, \langle y_\mathcal{C}, x\rangle)$ with probability at least $1/2 + \varepsilon$. Then, there is an extractor $(\mathcal{A}', \mathcal{B}', \mathcal{C}')$ that given input (ρ, σ) outputs (x, x) (running $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ as a subprotocol) with probability $\mathsf{poly}(\varepsilon)$.*

▶ Remark 31. Lemma 30 proves a special case of the simultaneous inner product conjecture postulated in [3]. Roughly speaking, the simultaneous inner product conjecture states that if a set of bipartite states $\{\rho_x\}_{x \in \{0,1\}^n}$ is such that any non-local adversary $(\mathcal{B}, \mathcal{C})$ given ρ_x, where $x \xleftarrow{\$} \{0,1\}^n$, cannot recover x (except with negligible probability) then $\mathcal{B}$ and $\mathcal{C}$ cannot distinguish Goldreich-Levin samples versus uniform samples (except with negligible advantage). The conjecture is parameterized by the distribution of the samples and also by the algebraic field associated with the samples. Lemma 30 shows that the conjecture is true for a correlated distribution of samples and when the field in question is $\mathbb{F}_2$.

▶ Remark 32. Lemma 30 resolves an important technical issue we will face in our unclonable security proof, similar to the one faced by [44][15]. Namely, $\mathcal{B}$ and $\mathcal{C}$ seem to get additional information about the hidden values $\langle y_\mathcal{B}, x\rangle, \langle y_\mathcal{C}, x\rangle$ by holding secret shares of their XOR value. Above we show that this is in fact not the case, i.e. the adversary does not get additional power from these shares. In [44], the authors utilized alternative security definitions to overcome this issue.

▶ **Theorem 33.** $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *above is secure against identical ciphertexts.*

***t*-Copy Security.** We show that our construction remains secure if up to $O(n)$ copies of the quantum ciphertext is given to the adversary in the unclonable security experiment.

[14] Here ρ is given to $\mathcal{A}$ and then a bipartite state σ is given to $\mathcal{B}$, $\mathcal{C}$ in the challenge phase.
[15] See Remark 7 (pp. 53) in [44].

▶ **Theorem 34.** *Let $c > 0$ be a constant and $t \leq (\log_2(1/\sqrt{C}) - c)n$, then* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *above is t-copy secure against identical ciphertexts.*

By instantiating $(\mathsf{Gen}_{\mathsf{UE}}, \mathsf{Enc}_{\mathsf{UE}}, \mathsf{Dec}_{\mathsf{UE}})$ with the construction of [17], we can set $C = 0.86$ and thus $t = n/10$. Therefore, by setting $n = 10t$ we get the following corollary:

▶ **Corollary 35.** *For any $t = \mathsf{poly}(\lambda)$, there exists a single-decryptor encryption scheme with quantum ciphertexts t-copy secure against identical ciphertexts (Definition 28) in the plain model.*

▶ Remark 36 (Bounded Number of Copies). Our construction is not t-copy secure against identical ciphertexts if t is an unbounded polynomial due to a simple attack that measures each copy of the ciphertext in the computational basis. Every measurement (except the first) will give a random linear constraint on the bits of x, hence a linear number of copies on average suffice to solve for x entirely. Nonetheless, we can set n accordingly for any fixed polynomial t.

4.2 Unclonable Encryption with Quantum Keys

Using our ideas in Section 4.1, we construct unclonable encryption with quantum keys in the plain model. Besides allowing the decryption key to be quantum, we do not relax the syntax of unclonable encryption. In particular, we achieve identical-challenge security with quantum challenges. Due to its similarity with Section 4.1, we leave the details of this section for the full version.

4.3 Secret Sharing

Definition. Below we formally define a quantum secret sharing scheme for a single-bit message $m \in \{0,1\}$, which is secure against an eavesdropper if only classical strings are leaked from the quantum shares.

▶ **Definition 37** (2-out-of-n Classical-Leakage-Resilient Quantum Secret Sharing)**.** *Let $n = \mathsf{poly}(\lambda)$. A* 2-out-of-$n$ classical-leakage-resilient quantum secret sharing scheme *is a tuple of algorithms* $(\mathsf{Share}, \mathsf{Rec})$ *with the following syntax:*

- $\mathsf{Share}(1^\lambda, m)$ *takes as input a security parameter and a message $m \in \{0,1\}$; it outputs a product state $|\psi_m\rangle = \bigotimes_{i=1}^n |\psi_m^i\rangle$ over registers $S_1, S_2, \ldots, S_n$ with $\mathsf{poly}(\lambda)$ qubits each.*
- $\mathsf{Rec}(i, j, \psi^i, \psi^j)$ *takes as input two indices $i, j \in [n]$ and quantum shares $|\psi^i\rangle, |\psi^j\rangle$ over registers S_i, S_j. It outputs a message $m \in \{0,1\}$.*

It satisfies the following properties:

1. **δ-Correctness:** *For all $m \in \{0,1\}, \lambda \in \mathbb{N}, (i,j) \in [n] \times [n]$ we have*

$$\Pr\left[m \leftarrow \mathsf{Rec}(i, j, \psi_m^i, \psi_m^j) \ : \ |\psi\rangle = \bigotimes_{i=1}^n |\psi_m^i\rangle \leftarrow \mathsf{Share}(1^\lambda, m)\right] \geq \delta.$$

2. **Perfect Secrecy:** *For all $i \in [n]$, we have*

$$\mathop{\mathbb{E}}_{|\psi_0\rangle \leftarrow \mathsf{Share}(1^\lambda, 0)} |\psi_0^i\rangle\langle\psi_0^i| = \mathop{\mathbb{E}}_{|\psi_1\rangle \leftarrow \mathsf{Share}(1^\lambda, 1)} |\psi_1^i\rangle\langle\psi_1^i|.$$

3. **ℓ-bit ε Classical-Leakage-Resilience:** *For any quantum algorithms* $\mathsf{Leak}_1, \ldots, \mathsf{Leak}_n$ *that output* ℓ *classical bits, any* n*-partite state* ρ*, and any distinguisher* $\mathcal{A}$*, we have*

$$\left| \Pr\left[1 \leftarrow \mathcal{A}(y_1, \ldots, y_n) \ : \ \begin{smallmatrix} \psi_0 \leftarrow \mathsf{Share}(1^\lambda, 0) \\ y_i \leftarrow \mathsf{Leak}_i(\psi_0^i, \rho_i),\ i \in [n] \end{smallmatrix}\right] - \Pr\left[1 \leftarrow \mathcal{A}(y_1, \ldots, y_n) \ : \ \begin{smallmatrix} \psi_1 \leftarrow \mathsf{Share}(1^\lambda, 1) \\ y_i \leftarrow \mathsf{Leak}_i(\psi_1^i, \rho_i),\ i \in [n] \end{smallmatrix}\right] \right| \leq \varepsilon,$$

where ρ_i *is the* i*-th register of* ρ*.*

▶ Remark 38. Note that we consider the perfect secrecy and the classical leakage resilience properties separately. It is natural to ask if it is possible to combine the two properties into a stronger property that guarantees security even against adversaries who in addition to receiving one of the shares, also receives as input classical leakage on the rest of the shares. Unfortunately, this stronger property cannot be satisfied due to a simple attack via quantum teleportation. Thus, we need to consider these two properties separately.

▶ Remark 39. The above notion can be generalized in many ways. Firstly, for simplicity, we define the shares to be pure states and we could consider a general notion where the shares could be entangled with each other. Secondly, we can consider sharing schemes guaranteeing security against different adversarial access structures.

4.3.1 Construction

We will construct the primitive above using the Haar distribution $\mathcal{D} = \mathcal{H}_d$ for $d = 2^{4\ell+\lambda}$. This satisfies $(\varepsilon, \mathcal{D}^t, \ell)$-SSI for $\varepsilon = \mathsf{negl}(\lambda)$ by Corollary 14 and Theorem 22. $\mathcal{H}_d$ can in turn be instantiated using a nt-design. We will pick $\omega(\log \lambda) \leq t \leq \mathsf{poly}(\lambda)$, so that the construction is efficient.

- $\mathsf{Share}(1^\lambda, m)$: Set $d = 2^{4n\ell+\lambda}$ and $\mathcal{D} = \mathcal{H}_d$. If $m = 0$, it samples nt copies of $|\psi\rangle \leftarrow \mathcal{D}$ and outputs $|\psi\rangle_{S_1}^{\otimes t} \otimes \ldots \otimes |\psi\rangle_{S_n}^{\otimes t}$. If $m = 1$, it independently samples t copies of $|\psi_i\rangle \leftarrow \mathcal{D}$ for $i \in [n]$ and outputs $|\psi_1\rangle_{S_1}^{\otimes t} \otimes \ldots \otimes |\psi_n\rangle_{S_n}^{\otimes t}$.
- $\mathsf{Rec}(i, j, |\varphi\rangle_{S_i}, |\varphi'\rangle_{S_j})$: It parses $S_i = S_i^{(1)} \ldots S_i^{(t)}$ and $S_j = S_j^{(1)} \ldots S_i^{(t)}$ as t registers. It applies a SWAP test to the registers $S_i^{(k)}$ and $S_j^{(k)}$ for $k \in [t]$. It outputs 0 if at least $\lfloor 3t/4 \rfloor$ of the SWAP tests succeed and outputs 1 otherwise.

4.3.2 Correctness and Secrecy

The construction above has δ-correctness for $\delta \geq 1 - \mathsf{negl}(\lambda)$. Note that for $m = 0$ each SWAP test will succeed with probability 1, so Rec will always output the correct message. If $m = 1$ on the other hand, each SWAP test will succeed with probability negligibly close to $1/2$, so by a Chernoff bound Rec will output 0 with only negligible probability since $t = \omega(\log \lambda)$.

Perfect secrecy is trivial given that each share is distributed according to $\mathcal{D}$.

4.3.3 Classical Leakage Resilience

By Corollary 14 and Theorem 22, the total variation distance between the output distributions of any ℓ-bit leakage functions $\mathsf{Leak}_1, \ldots, \mathsf{Leak}_n$ with respect to shares of $m = 0$ and $m = 1$ is at most

$$O\left(\frac{2^{2n\ell} n^3 t^2}{\sqrt{d}}\right) = O\left(\frac{n^3 t^2}{2^{\lambda/2}}\right) \leq \mathsf{negl}(\lambda).$$

References

1 Scott Aaronson. Quantum copy-protection and quantum money. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 229–242. IEEE, 2009. doi:10.1109/CCC.2009.42.

2 Divesh Aggarwal, Ivan Damgård, Jesper Buus Nielsen, Maciej Obremski, Erick Purwanto, Joao Ribeiro, and Mark Simkin. Stronger leakage-resilient and non-malleable secret sharing schemes for general access structures. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39*, pages 510–539. Springer, 2019. doi:10.1007/978-3-030-26951-7_18.

3 Prabhanjan Ananth and Amit Behera. A modular approach to unclonable cryptography. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024*, pages 3–37, Cham, 2024. Springer Nature Switzerland. doi:10.1007/978-3-031-68394-7_1.

4 Prabhanjan Ananth, Aditya Gulati, and Yao-Ting Lin. Cryptography in the common haar state model: Feasibility results and separations. In Elette Boyle and Mohammad Mahmoody, editors, *Theory of Cryptography*, pages 94–125, Cham, 2025. Springer Nature Switzerland.

5 Prabhanjan Ananth and Fatih Kaleoglu. Unclonable encryption, revisited. In *Theory of Cryptography Conference*, pages 299–329. Springer, 2021. doi:10.1007/978-3-030-90459-3_11.

6 Prabhanjan Ananth, Fatih Kaleoglu, Xingjian Li, Qipeng Liu, and Mark Zhandry. On the feasibility of unclonable encryption, and more. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022*, pages 212–241, Cham, 2022. Springer Nature Switzerland. doi:10.1007/978-3-031-15979-4_8.

7 Prabhanjan Ananth, Fatih Kaleoglu, and Qipeng Liu. Cloning games: A general framework for unclonable primitives. In *CRYPTO*, 2023.

8 Prabhanjan Ananth, Fatih Kaleoglu, and Henry Yuen. Simultaneous haar indistinguishability with applications to unclonable cryptography, 2024. arXiv:2405.10274, doi:10.48550/arXiv.2405.10274.

9 Prabhanjan Ananth and Rolando L. La Placa. Secure software leasing. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, pages 501–530, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-77886-6_17.

10 Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im) possibility of obfuscating programs. In *Annual international cryptology conference*, pages 1–18. Springer, 2001. doi:10.1007/3-540-44647-8_1.

11 James Bartusek, Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Obfuscation of pseudo-deterministic quantum circuits. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1567–1578, 2023. doi:10.1145/3564246.3585179.

12 James Bartusek and Giulio Malavolta. Indistinguishability obfuscation of null quantum circuits and applications. In *ITCS*, 2022.

13 Fabrice Benhamouda, Akshay Degwekar, Yuval Ishai, and Tal Rabin. On the local leakage resilience of linear secret sharing schemes. *Journal of Cryptology*, 34:1–65, 2021.

14 Charles H Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Physical review letters*, 70(13):1895, 1993.

15 Charles H Bennett, David P DiVincenzo, Christopher A Fuchs, Tal Mor, Eric Rains, Peter W Shor, John A Smolin, and William K Wootters. Quantum nonlocality without entanglement. *Physical Review A*, 59(2):1070, 1999.

16 Anne Broadbent and Raza Ali Kazmi. Constructions for quantum indistinguishability obfuscation. In *International Conference on Cryptology and Information Security in Latin America*, pages 24–43. Springer, 2021. doi:10.1007/978-3-030-88238-9_2.

17 Anne Broadbent and Sébastien Lord. Uncloneable Quantum Encryption via Oracles. In Steven T. Flammia, editor, *15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020)*, volume 158 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:22, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.TQC.2020.4.

18 Francesco Buscemi. All entangled quantum states are nonlocal. *Physical review letters*, 108(20):200401, 2012.

19 Alper Çakan and Vipul Goyal. Unclonable cryptography with unbounded collusions. *Cryptology ePrint Archive*, 2023.

20 Alper Cakan, Vipul Goyal, Chen-Da Liu-Zhang, and João Ribeiro. Unbounded leakage-resilience and intrusion-detection in a quantum world. *Cryptology ePrint Archive*, 2023.

21 Eshan Chattopadhyay, Jesse Goodman, Vipul Goyal, Ashutosh Kumar, Xin Li, Raghu Meka, and David Zuckerman. Extractors and secret sharing against bounded collusion protocols. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1226–1242. IEEE, 2020. doi:10.1109/FOCS46700.2020.00117.

22 Sitan Chen, Jordan Cotler, Hsin-Yuan Huang, and Jerry Li. Exponential separations between learning with and without quantum memory. In *FOCS*, 2021.

23 Andrew M Childs, Debbie Leung, Laura Mančinska, and Maris Ozols. A framework for bounding nonlocality of state discrimination. *Communications in Mathematical Physics*, 323:1121–1153, 2013.

24 Eric Chitambar and Min-Hsiu Hsieh. Asymptotic state discrimination and a strict hierarchy in distinguishability norms. *Journal of Mathematical Physics*, 55(11), 2014.

25 Eric Chitambar, Debbie Leung, Laura Mančinska, Maris Ozols, and Andreas Winter. Everything you always wanted to know about locc (but were afraid to ask). *Communications in Mathematical Physics*, 328:303–326, 2014.

26 Andrea Coladangelo and Sam Gunn. How to use quantum indistinguishability obfuscation. In *STOC*, 2024.

27 Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. Hidden cosets and applications to unclonable cryptography. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, pages 556–584, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-84242-0_20.

28 Andrea Coladangelo, Christian Majenz, and Alexander Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model. *Quantum*, 8:1330, May 2024. doi:10.22331/q-2024-05-02-1330.

29 David P DiVincenzo, Debbie W Leung, and Barbara M Terhal. Quantum data hiding. *IEEE Transactions on Information Theory*, 48(3):580–598, 2002. doi:10.1109/18.985948.

30 Llorenç Escolà-Farràs, Jaròn Has, Maris Ozols, Christian Schaffner, and Mehrdad Tahmasbi. Parallel repetition of local simultaneous state discrimination. *arXiv preprint arXiv:2211.06456*, 2022.

31 Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM Journal on Computing*, 45(3):882–929, 2016. doi:10.1137/14095772X.

32 Julio Gea-Banacloche. Hiding messages in quantum data. *Journal of Mathematical Physics*, 43(9):4531–4536, 2002.

33 Marios Georgiou and Mark Zhandry. Unclonable decryption keys. *IACR Cryptol. ePrint Arch.*, 2020:877, 2020. URL: https://eprint.iacr.org/2020/877.

34 O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, pages 25–32, New York, NY, USA, 1989. Association for Computing Machinery. doi:10.1145/73007.73010.

35 Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 685–698, 2018. doi:10.1145/3188745.3188872.

36 Saronath Halder, Manik Banik, Sristy Agrawal, and Somshubhro Bandyopadhyay. Strong quantum nonlocality without entanglement. *Physical review letters*, 122(4):040403, 2019.

37 Aram W. Harrow. The church of the symmetric subspace, 2013. arXiv:1308.6595.

38 Aram W. Harrow. Approximate orthogonality of permutation operators, with application to quantum information, 2023. arXiv:2309.00715.

39 Patrick Hayden, Debbie Leung, and Graeme Smith. Multiparty data hiding of quantum information. *Physical Review A*, 71(6):062339, 2005.

40 Carl W Helstrom. Quantum detection and estimation theory. *Journal of Statistical Physics*, 1:231–252, 1969.

41 Taiga Hiroka, Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Robust combiners and universal constructions for quantum cryptography. *arXiv preprint arXiv:2311.09487*, 2023.

42 Yael Tauman Kalai and Leonid Reyzin. A survey of leakage-resilient cryptography. In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pages 727–794. Association for Computing Machinery, 2019. doi:10.1145/3335741.3335768.

43 Fuyuki Kitagawa and Ryo Nishimaki. One-out-of-many unclonable cryptography: Definitions, constructions, and more. In *Theory of Cryptography Conference*, pages 246–275. Springer, 2023. doi:10.1007/978-3-031-48624-1_10.

44 Srijita Kundu and Ernest Y. Z. Tan. Device-independent uncloneable encryption, 2022. doi:10.48550/arXiv.2210.01058.

45 Christian Majenz, Maris Ozols, Christian Schaffner, and Mehrdad Tahmasbi. Local simultaneous state discrimination, 2021. arXiv:2111.01209.

46 William Matthews, Stephanie Wehner, and Andreas Winter. Distinguishability of quantum states under restricted families of measurements with an application to quantum data hiding. *Communications in Mathematical Physics*, 291:813–843, 2009.

47 Antonio Anna Mele. Introduction to haar measure tools in quantum information: A beginner's tutorial, 2024. arXiv:2307.08956.

48 Michael Nathanson. Distinguishing bipartitite orthogonal states using locc: Best and worst cases. *Journal of Mathematical Physics*, 46(6), 2005.

49 Roger Penrose. Applications of negative dimensional tensors. *Welsh, D., Ed., Combinatorial Mathematics and Its Applications*, pages 221–244, 1971.

50 Asher Peres and William K Wootters. Optimal detection of quantum information. *Physical Review Letters*, 66(9):1119, 1991.

51 Marco Piani, Varun Narasimhachar, and John Calsamiglia. Quantumness of correlations, quantumness of ensembles and quantum data hiding. *New Journal of Physics*, 16(11):113001, 2014.

52 Robert Raussendorf and Tzu-Chieh Wei. Quantum computation by local measurement. *Annual Review of Condensed Matter Physics*, 3(Volume 3, 2012):239–261, 2012. doi:10.1146/annurev-conmatphys-020911-125041.

53 Akshayaram Srinivasan and Prashant Nalini Vasudevan. Leakage resilient secret sharing and applications. In *Annual International Cryptology Conference*, pages 480–509. Springer, 2019. doi:10.1007/978-3-030-26951-7_17.

54 Stephen Wiesner. Conjugate coding. *ACM Sigact News*, 15(1):78–88, 1983. doi:10.1145/1008908.1008920.

55 Mark Zhandry. Quantum lightning never strikes the same state twice. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 408–438, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-17659-4_14.

Single-Round Proofs of Quantumness from Knowledge Assumptions

Petia Arabadjieva
Institute for Theoretical Physics, ETH Zurich, Switzerland

Alexandru Gheorghiu
Department of Computer Science and Engineering, Chalmers University of Technology, Göteborg, Sweden

Victor Gitton
Institute for Theoretical Physics, ETH Zurich, Switzerland

Tony Metger
Institute for Theoretical Physics, ETH Zurich, Switzerland

Abstract

A *proof of quantumness* is an efficiently verifiable interactive test that an efficient quantum computer can pass, but all efficient classical computers cannot (under some cryptographic assumption). Such protocols play a crucial role in the certification of quantum devices. Existing single-round protocols based solely on a cryptographic hardness assumption (like asking the quantum computer to factor a large number) require large quantum circuits, whereas multi-round ones use smaller circuits but require experimentally challenging mid-circuit measurements.

In this work, we construct efficient single-round proofs of quantumness based on existing *knowledge assumptions*. While knowledge assumptions have not been previously considered in this context, we show that they provide a natural basis for separating classical and quantum computation. Our work also helps in understanding the interplay between black-box/white-box reductions and cryptographic assumptions in the design of proofs of quantumness. Specifically, we show that multi-round protocols based on Decisional Diffie-Hellman (DDH) or Learning With Errors (LWE) can be "compiled" into single-round protocols using a knowledge-of-exponent assumption [7] or knowledge-of-lattice-point assumption [36], respectively. We also prove an *adaptive hardcore-bit* statement for a family of claw-free functions based on DDH, which might be of independent interest.

2012 ACM Subject Classification Theory of computation → Cryptographic protocols

Keywords and phrases Proofs of quantumness, Knowledge assumptions, Learning with errors, Decisional Diffie-Hellman

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.8

Related Version *Full Version*: `https://arxiv.org/abs/2405.15736` [5]

Funding P.A., V.G. and T.M. acknowledge support from the ETH Zurich Quantum Center and the Air Force Office for Scientific Research, grant No. FA9550-19-1-0202.
Alexandru Gheorghiu: Knut and Alice Wallenberg Foundation, through the Wallenberg Centre for Quantum Technology (WACQT).
Tony Metger: ETH Doc.Mobility Fellowship.

Acknowledgements We thank Alex Lombardi, Urmila Mahadev, Greg Kahanamoku-Meyer, Umesh Vazirani, John Wright, and Tina Zhang for helpful discussions. We are especially grateful to Vinod Vaikuntanathan to suggesting many of these ideas in the early stages of the project.

1 Introduction

Demonstrating quantum advantage, the point where a quantum computer can solve a problem that no existing classical computer can, is both a theoretical and technological challenge. It requires a problem that is plausibly intractable for classical algorithms and which admits an


© Petia Arabadjieva, Alexandru Gheorghiu, Victor Gitton, and Tony Metger;
licensed under Creative Commons License CC-BY 4.0



efficient quantum algorithm, ideally one that can be performed with noisy intermediate-scale quantum (NISQ) devices [42]. Additionally, the problem's solution should be efficiently verifiable by a classical computer. This is necessary if one wishes to have a convincing and scalable way of proving quantum advantage. Currently, there are three main paradigms for demonstrating quantum advantage.

The most straightforward one is to solve a problem which is believed to be classically hard, such as integer factorization. A quantum computer could efficiently solve this task by running Shor's algorithm [43] and the solution can be efficiently verified by multiplying the factors reported by the prover. But while Shor's algorithm is efficient in the sense of only requiring a polynomial-size quantum circuit, the actual circuit for any reasonably-sized integer to be factored is too large to implement with NISQ devices [22, 25]. Another approach is based on the classical hardness of sampling problems such as *random circuit sampling* [12, 6, 44] or *boson sampling* [1, 48, 37]. While these experiments can be implemented with current hardware, they are not efficiently verifiable.

The work of Brakerski et al. [13] introduced a new approach towards testing for quantum advantage, referred to as a *proof of quantumness* protocol. Akin to the cryptographic notions of proof or argument systems [24, 15, 23], a proof of quantumness is an interactive protocol between a polynomial-time classical *verifier* and an ostensibly quantum polynomial-time *prover*. The verifier issues challenges to the prover and checks the correctness of the prover's responses. The key feature of such a protocol is that there should exist an efficient quantum strategy that allows the prover to correctly answer the verifier's challenges with high probability, whereas any efficient classical strategy can only succeed with low probability (under some plausible cryptographic hardness assumption such as the classical intractability of factoring, Decisional Diffie-Hellman (DDH), or the Learning With Errors (LWE) problem).

In contrast to other paradigms for proving quantum advantage, the cryptographic proofs of quantumness of Brakerski et al. and follow-up works do not require the quantum prover to break the underlying computational hardness assumption. Instead, they leverage the fact that the restriction imposed on the prover through mid-protocol interaction limits a classical prover's capacity for correctly responding to subsequent challenges from the verifier more strongly than it limits a quantum prover.

The advantage of these protocols over an integer factorization-based test is that they have much smaller circuits than Shor's algorithm, making them potentially more suitable for implementation on near-term devices [30, 29, 49, 26, 35, 4]. However, due to their interactive nature, they require the honest quantum prover to perform mid-circuit measurements for each of the verifier's challenges. Mid-circuit measurements on a subset of qubits are difficult to implement on existing quantum devices without disturbing neighboring qubits and can thus degrade the quality of the remaining computation. An experimental implementation of the two-round protocol by Brakerski et al. [49] directly compared the performance of an ion-trap quantum computer running the protocol with and without mid-circuit measurements, revealing a significant difference. Thus, implementing this proof of quantumness protocol at the scale required for quantum advantage seems especially challenging.[1] It would therefore be desirable to have proofs of quantumness with relatively small quantum circuits and which involve only one round[2] of interaction between the verifier and the prover. Beyond the practical motivation, this would also give us a better understanding of the structure required for demonstrating quantum advantage in a way that is efficiently verifiable.

[1] Note that the instance sizes of the underlying cryptographic problem in [49] are so small that they can easily be broken by a laptop. For a convincing demonstration of quantum advantage, one would have to use instance sizes that cannot be broken even by the fastest classical supercomputers.

[2] Throughout the paper, we use the convention that a one-round protocol refers to a protocol with two messages, one message from the verifier to the prover and one message back.

The recent breakthrough work of Yamakawa and Zhandry made progress in this direction by giving a single-round proof of quantumness protocol in the *random oracle model* (ROM) [45]. Prior to their work, Brakerski et al. constructed single-round proofs of quantumness in the ROM that additionally required a structured computational assumption, such as factoring or LWE [14]. However, with both of these approaches the circuits that the prover would have to perform are possibly larger than those in existing multi-round proofs of quantumness [13, 29].

Our main result is a single-round proof of quantumness that only requires the same small circuits as an existing multi-round interactive proof of quantumness and is based on *knowledge assumptions*, a cryptographic idea that turns out to be natural for proofs of quantumness. We achieve this by starting from the two-round protocol of [13] and removing one of the rounds of interaction through the use of a knowledge assumption [17, 40, 16, 36, 32]. As explained in [32], a knowledge assumption is a statement of the form:

> "If an algorithm $\mathcal{A}$ outputs an object of type X, it must know a corresponding witness of the type W, such that the output and the witness are in some relation $\mathcal{R} \subseteq X \times W$."

The rationale behind knowledge assumptions is that certain computational tasks, performed by some probabilistic algorithm $\mathcal{A}$, can only be performed efficiently by following a specific sequence of steps, thus obtaining a series of intermediate values. Informally, we say that $\mathcal{A}$ must have "known" the intermediate values for its specific output. This is made more precise by saying that there exists an efficient *extractor* $\mathcal{A}^*$ that receives as input $\mathcal{A}$'s random coins and outputs (or *extracts*) the relevant intermediate values of $\mathcal{A}$.

Knowledge assumptions have traditionally been used for the design of protocols that require both extractability (like in proof/argument of knowledge protocols) and succinctness [9, 32]. In our case, we are able to leverage knowledge assumptions to construct efficient, single-round proofs of quantumness. As far as we are aware, this is the first time knowledge assumptions are used in this context. We argue that their application here is natural and in some sense necessary, if one wishes to avoid multi-round interaction or working in the ROM. Intuitively, this is because some aspect of the proof of quantumness protocol has to differentiate between classical and quantum provers. In multi-round protocols, this distinction comes from the fact that classical provers can be *rewound*, but quantum provers generally cannot. In a ROM-based protocol, the distinction arises from the ability to *record* classical queries to a random oracle in a way that is not possible quantumly.

In our single-round protocols, the classical-quantum distinction is due to the knowledge assumption: for example, if $f : \mathcal{X} \to \mathcal{Y}$ is a one-way function with a sparse range (i.e. most values in $\mathcal{Y}$ are not in the range of f), it is plausible that a *classical* prover that produces a value in the range of f must have done so by evalating the function on some $x \in \mathcal{X}$, and must therefore also "know" the corresponding preimage; this can be captured formally as a knowledge assumption [16, 8]. However, a *quantum* prover can do the following: first prepare the state $\sum_x |x\rangle|f(x)\rangle$, then measure the second register in the computational basis to obtain an image $y \in \text{range}(f)$, and measure the first register in the Hadamard basis to "erase" any knowledge of the preimage x. Such a prover can be said to produce a value in the range of f without knowledge of a preimage. It is this distinction between classical and quantum computation that our single-round proofs of quantumness exploit. We note that a related idea of oblivious LWE sampling was recently explored in [18]; since a proof of quantumness only requires soundness against classical provers, we do not need to analyse the quantum prover in detail, but it might be interesting to explore the relation between knowledge assumptions used in our work and the (strong) notion of oblivious LWE sampling proposed in [18]. We discuss the classical-quantum distinction and the impossibility of single-round proofs of quantumness with black-box security reductions in more detail in Section 3.

Organsation

The rest of this work is structured as follows: in Section 1.1, we give a brief overview of the two-round proof of quantumness from [13]. This will form the basis for our single-round proof of quantumness. In Section 1.2, we recall two existing knowledge assumptions from [7, 36] that we use in our protocols. In Section 2, we explain how to make use of these knowledge assumptions to obtain different single-round proofs of quantumness. In Section 3 we argue that a white-box assumption (like a knowledge assumption) seems to be necessary for single-round proofs of quantumness. Finally, in Section 4 and Section 5 we discuss additional related works and open questions. For a formal description of our results and proofs, we refer to the full version of this paper [5].

1.1 A two-round proof of quantumness

To explain our results, we first need to outline the two-round (four-message) proof of quantumness protocol from [13]. At the heart of this protocol is a collection of functions known as Trapdoor Claw-free Functions (TCF). TCFs are a type of collision-resistant hash function – they are 2-to-1 functions for which it should be intractable to find a colliding pair of inputs (known as a *claw*), given the description of the function. Additionally, the functions are generated with a trapdoor that allows for efficient inversion. The specific TCFs used in [13] require an additional property known as the *adaptive hardcore-bit* (AHCB) property. Informally, this states that it is not only intractable to find collisions, but given any particular input x_0 and corresponding image under the TCF, denoted $y = f(x_0)$, it should be intractable to recover even a single bit of x_1, the other input with which x_0 forms a claw (i.e. $f(x_1) = f(x_0) = y$). Prior to our work, constructing TCFs with an AHCB had only been achieved from LWE and (non-standard) hardness assumptions of isogeny-based group actions [3]. One of our results shows an AHCB property for a TCF based on DDH.

Algorithm 1 The Proof of Quantumness of [13] (informal).

1. The verifier generates a description of a TCF f, together with its trapdoor, t. It then sends f to the prover.
2. The prover sends the verifier a point y in the image of f. Denote as x_0 and x_1 the associated preimages, so $y = f(x_0) = f(x_1)$.
3. With probability 1/2, the verifier sends the prover one of the following two challenges:
   - **(a) Preimage test.** The verifier asks the prover for a valid preimage of y. Denoting the prover's response as x, the verifier accepts iff $f(x) = y$.
   - **(b) Equation test.** The verifier asks the prover for a non-zero string d, such that $d \cdot (x_0 \oplus x_1) = 0$. The verifier uses the trapdoor, t, to recover x_0 and x_1 from y and then checks whether the equation is satisfied, accepting if it is and rejecting otherwise.

Given a family of TCFs, with the AHCB property, the Brakerski et al. protocol from [13] is described informally in Protocol 1. Brakerski et al. showed that, assuming the AHCB property, no polynomial-time classical prover makes the verifier accept with probability non-negligibly[3] larger than 3/4 (this is referred to as the *soundness* of the protocol). At the same time, they gave a simple quantum strategy which would allow the prover to succeed with probability 1 (referred to as the *completeness* of the protocol). This honest prover first creates an equal superposition over evaluations of f,

[3] A *negligible* function f, denoted by $f = \mathrm{negl}(\lambda)$, is a function $f : \mathbb{N} \to \mathbb{R}$ such that $f(\lambda) = o(\lambda^{-c})$ for every constant $c \in \mathbb{R}$.

$$\sum_x |x\rangle|f(x)\rangle\,.$$

The prover then measures the second register, resulting in a value $y = f(x_0) = f(x_1)$, and collapsing the state in the first register to

$$\frac{|x_0\rangle + |x_1\rangle}{\sqrt{2}}\,. \tag{1.1}$$

We can see that if this state is measured in the computational basis, it results in one of the two preimages of y, thus providing a valid response to the preimage test. Conversely, if the state is measured in the Hadamard basis (i.e. applying Hadamard gates to all qubits and measuring in the computational basis), the result will be a string d such that $d \cdot (x_0 \oplus x_1) = 0$, yielding a valid response to the equation test. Thus, a quantum prover implementing this strategy would make the verifier accept with probability 1.

The intuition for why it is classically intractable to succeed in this protocol is that a classical prover that answers both the preimage and equation tests correctly can be *rewound* so as to obtain both a valid preimage and a valid equation. However, having both would contradict the AHCB property. In contrast, a quantum prover cannot be rewound and can use the state from Equation (1.1) to answer *either of the two challenges* correctly, but not both at the same time. Communication between the two rounds is crucial for the security proof, as a reduction to the AHCB property requires that the prover holds a preimage and equation *for the same* y. This can only be guaranteed if the prover commits to a fixed choice of y before receiving the second challenge.

A natural question is whether the equation test on its own is already classically intractable. Unfortunately, simply removing the preimage test breaks the security proof of [13]: the AHCB property says that it is hard to produce an image, an equation, *and a preimage* together. Therefore, without the preimage test one cannot use a successful classical prover in the proof of quantumness to break the AHCB property, as one does not have access to a preimage[4]. Indeed, it can be shown that no *black-box* reduction can reduce the security of this "equation-test only" proof of quantumness to LWE ([39], see also Section 3).

Our results show that a variation of the "equation-test only" proof of quantumness can be made to work, provided we use a knowledge assumption to replace the role of the preimage test. Intuitively, the knowledge assumption can be used to "extract" a preimage from a successful classical prover without the need for an explicit preimage test. Since a knowledge assumption deals with the inner workings of a prover, our result can be interpreted as a *white-box* reduction from a version of the equation test to the AHCB property.

1.2 Knowledge assumptions

Knowledge assumptions posit the existence of an efficient extractor that is able to produce certain intermediate values that are in some relation with the output of an algorithm. A canonical example is the so-called *knowledge of exponent assumption* (KEA), introduced by Damgård in [17]. Informally, this says that given a generator g of some multiplicative

[4] We remark here that this is actually not the only problem that arises when removing the preimage test. In fact, Urmila Mahadev observed that for both constructions we consider, there exists a classical winning strategy in the equation test which involves evaluating f on an extended domain. In the original Brakerski et al. protocol [13], this strategy is excluded by the preimage test. In our protocols, this strategy is excluded by adding another type of test which does not increase the number of rounds of the protocol and is explained in Section 2.

group $\mathbb{G}$, as well g^α for some random power α, the only way to produce a new pair of the form (h, h^α) is by *exponentiating* the original pair. In other words, any efficient (randomized) algorithm $\mathcal{A}$ which outputs (h, h^α) given (g, g^α) must "know" an exponent k such that $h = g^k$. This is formalized by saying that for every such algorithm $\mathcal{A}$, there exists an efficient extractor $\mathcal{A}^*$ that, given (g, g^α) and the random coins of $\mathcal{A}$ for which $\mathcal{A}$ outputs (h, h^α), will output (or *extract*) the exponent k such that $h = g^k$ with overwhelming probability:

▶ **Assumption 1** (KEA (informal), from [17]). *Let $\mathcal{A}$ be an efficient probabilistic algorithm which receives as input a generator g of a multiplicative group $\mathbb{G}$ and a random group element g^α. Then there exists an efficient extractor $\mathcal{A}^*$, which uses the same input and random coins r as $\mathcal{A}$, and such that*

$$\Pr\left[\begin{array}{c}(h,h') \leftarrow \mathcal{A}(g, g^\alpha, r) \\ h^\alpha = h'\end{array} \wedge \begin{array}{c} k \leftarrow \mathcal{A}^*(g, g^\alpha, r) \\ g^k \neq h\end{array}\right] = \mathrm{negl}(\lambda)\,. \tag{1.2}$$

Variations of this assumption have also been considered, such as the t-KEA, in [7], in which the input to $\mathcal{A}$ is instead of the form $(g^{\mathbf{r}}, g^{\alpha\mathbf{r}})$, where $\mathbf{r}$ is a t-dimensional vector and exponentiation is element-wise.

The rationale behind knowledge of exponent assumptions is that the function that maps k to $(g^k, g^{\alpha k})$ has a sparse image (considered as a subset of $\mathbb{G} \times \mathbb{G}$). Thus, it seems implausible that $\mathcal{A}$ could come up with a valid image simply by cleverly sampling its output from $\mathbb{G} \times \mathbb{G}$ without exponentiating the input.[5]

More recently, similar knowledge assumptions were proposed for lattices. One example, introduced in [36], is known as the *knowledge of lattice point assumption* (denoted as LK-ϵ). This says the following: suppose there is an efficient randomized classical algorithm $\mathcal{A}$ which takes as input a lattice basis $\mathbf{A}$ and outputs a point $\mathbf{c}$ that is ϵ-close to the lattice $\mathcal{L}(\mathbf{A})$. Then $\mathcal{A}$ must "know" the lattice point closest to $\mathbf{c}$.

▶ **Assumption 2** (LK-ϵ (informal), from [36]). *Let ϵ be a fixed constant in $(0, 1/2)$. Denote by $\lambda(\mathcal{L})$ the norm of the shortest vector in a lattice $\mathcal{L}$. Let $\mathcal{A}$ be an efficient probabilistic algorithm which receives as input a basis $\mathbf{A}$ of a lattice $\mathcal{L}$ and outputs a vector $\mathbf{y}$. Then there exists an efficient algorithm $\mathcal{A}^*$ which uses the same input and random coins r as $\mathcal{A}$, and outputs a lattice point $\mathbf{p} \in \mathcal{L}$ such that*

$$\Pr\left[\begin{array}{c}\mathbf{y} \leftarrow \mathcal{A}(\mathbf{A}, r) \\ \exists\ \mathbf{c} \in \mathcal{L}(\mathbf{A})\ s.t.\ ||\mathbf{c} - \mathbf{y}|| \leq \epsilon \cdot \lambda(\mathcal{L}(\mathbf{A}))\end{array} \wedge \begin{array}{c}\mathbf{p} \leftarrow \mathcal{A}^*(\mathbf{A}, r) \\ \mathbf{p} \neq \mathbf{c}\end{array}\right] = \mathrm{negl}(\lambda)\,. \tag{1.3}$$

The motivation for this assumption is similar to that of KEA – the set of points that are close to the lattice (for a suitable choice of ϵ) is sparse in the set of all points, and the only apparent efficient strategy to sample from this sparse set is to pick a random lattice point and perturb it. This lattice knowledge assumption and variants of it have been used to construct efficient FHE and SNARK schemes [19, 27].

For our results, we will use t-KEA and LK-ϵ, which we state formally in [5, Assumption 4] and [5, Assumption 5].

[5] An alternative way of coming up with an image would be if $\mathcal{A}$ could determine α, given g and g^α. However, this would entail solving the discrete logarithm problem, which is believed to be classically intractable.

2 Main results

Our main result is that combining knowledge assumptions with standard cryptographic assumptions, like LWE or DDH, leads to efficient single-round proof of quantumness protocols. To make our results modular, we first show how to construct a general single-round proof of quantumness from a cryptographic primitive that we call Doubly Extended Extractable Noisy Trapdoor Claw-free Functions (abbreviated e^{3}NTCF). Second, we give two constructions of e^{3}NTCF: one from the DDH and t-KEA assumptions, and one from the LWE and LK-ϵ assumptions.

2.1 Single-round proof of quantumness from e^{3}NTCF

Our starting point is the protocol from [13], which we explained in Section 1.1. Recall that there, one uses a TCF with the AHCB property, and argues that if a classical prover could succeed in the preimage and equation tests, by rewinding we could construct a tuple (x, y, d) of a preimage, image, and equation that would contradict the AHCB property.

Now suppose that the TCF family $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y}\}$ used in this protocol had an additional extractability property: for any classical prover that produces an image $y \in \text{range}(f)$, there exists an extractor that produces a corresponding preimage x. This is, in essence, a knowledge assumption. With such an extractable TCF, we could simply remove the preimage test from Protocol 1: then, if we had a classical prover that succeeded in this modified protocol, we could use that prover to find an image $y \in \text{range}(f)$ and equation d, and use the extractor to find a corresponding preimage x, such that (x, y, d) break the AHCB property.

Unfortunately, we do not know how to construct such an extractable TCF with AHCB from existing knowledge assumptions such as t-KEA or LK-ϵ[6]. The key difficulty here is that the extractability and AHCB properties have to hold simultaneously.

One way to circumvent this issue would be to introduce an additional function family $\mathcal{H} = \{h : \mathcal{X} \to \mathcal{Y}\}$ that is indistinguishable from the TCF family $\mathcal{F}$ (i.e. given a description of a random choice of either kind of function, it is hard to tell which kind of function it is). This function family $\mathcal{H}$ can be constructed such that it has an extractability property, i.e. if a classical algorithm produces a value y in the image of h, an extractor can find a preimage x (under a standard knowledge assumption such as t-KEA). However, $\mathcal{H}$ itself is not a TCF and has no AHCB property.

We now want to combine the AHCB property of $\mathcal{F}$ with the extractability property of $\mathcal{H}$. For this we leverage that the two function families are computationally hard to distinguish[7]: if we send a description of either $f \in \mathcal{F}$ or $h \in \mathcal{H}$ to a classical prover, the prover cannot tell which kind of function it received. This suggests the following protocol.

1. The verifier generates a description of a TCF $f \in \mathcal{F}$ (with a trapdoor t) or an extractable function $h \in \mathcal{H}$ and sends this function description to the prover.

[6] To be precise, we found that using these knowledge assumptions directly on the known NTCF constructions based on DDH and LWE does not work to infer extractability.

[7] One may ask here why we do not use computational indistinguishability to directly transfer the extractability property of $\mathcal{H}$ to $\mathcal{F}$, given that the extractor is an efficient algorithm. This has to do with the exact form of the extractability property: successful extraction is only guaranteed under the condition $y \in \text{range}(h)$, which cannot be checked efficiently without the trapdoor. Thus, while we can exploit the computational indistinguishability of $\mathcal{F}$ and $\mathcal{H}$ in our protocol, it is not possible to use computational indistinguishability to infer an analogous extractability property for $\mathcal{F}$.

2. The verifier receives (y, d) from the prover.
   - If the verifier sent a TCF f, it uses the trapdoor to recover x_0 and x_1 such that $y = f(x_0) = f(x_1)$ and accepts iff $d \cdot (x_0 \oplus x_1) = 0$.
   - If the verifier sent an extractable function h, it accepts iff $y \in \text{range}(h)$.

Suppose that a classical prover succeeded in this protocol with high probability. We can use this prover (and the corresponding extractor for the extractable function family) to break the AHCB property of f as follows: given a description of $f \in \mathcal{F}$, run the prover to generate (y, d). Then use the extractor on this prover on input $f \in \mathcal{F}$ to generate a preimage x. We claim that (x, y, d) violates the AHCB property.

Note that this reduction runs the extractor for the function family $\mathcal{H}$ on an input $f \in \mathcal{F}$, i.e. it runs the extractor on an input for which it was not designed. However, recall that descriptions of f and h are computationally indistinguishable. On the other hand, the extractor as well as the function that checks whether the extractor produced a correct preimage are efficient. Therefore, since the protocol ensures that on input h, the classical prover produces $y \in \text{range}(h)$, it follows that when we run the extractor for this classical prover on an input $f \in \mathcal{F}$, it will still produce a valid preimage of $y = f(x_0) = f(x_1)$; otherwise, we could distinguish f from h.

When trying to construct such a pair $(\mathcal{F}, \mathcal{H})$ of function families from DDH and t-KEA, we are faced with one additional technical challenge: the only evident construction of an extractable function family $\mathcal{H}$ works by extending the functions $f \in \mathcal{F}$ to a larger domain $\mathcal{X}'$ (on which $\mathcal{F}$ no longer satisfies the AHCB property) and constructing extractable functions $h : \mathcal{X}' \to \mathcal{Y}$, i.e. the extractability property of h only holds with respect to the extended domain $\mathcal{X}'$. This means that the extractor may, on input $y \in f(\mathcal{X})$, produce a $x \in \mathcal{X}' \setminus \mathcal{X}$ that also maps to y. While this is a valid preimage to y, it is not useful for breaking the AHCB property: for that we need a preimage $x \in \mathcal{X}$.

We therefore introduce a third function family, $\mathcal{G} = \{g : \mathcal{X}' \to \mathcal{Y}\}$, that is computationally indistinguishable from both $\mathcal{H}$ and the extension of $\mathcal{F}$. The functions in $\mathcal{G}$ are injective even on the larger domain $\mathcal{X}'$. In our proof of quantumness, the verifier will check that when sent a function g, the prover returns an image $y \in g(\mathcal{X})$, i.e. the image y must be in the range of the restricted domain $\mathcal{X}$. This essentially forces the prover to evaluate any function it receives only on the restricted domain[8] $\mathcal{X}$, since if it evaluated the *injective* function g on an input $x \in \mathcal{X}' \setminus \mathcal{X}$, the verifier would reject the resulting image y.

In summary, our single-round proof of quantumness relies on a triplet of function families $(\mathcal{F}, \mathcal{G}, \mathcal{H})$ with the following properties:[9]

▶ **Definition 1** (e^{3}NTCF, informal)**.** *A tuple of function families $(\mathcal{F}, \mathcal{G}, \mathcal{H})$ is called a* Doubly Extended Extractable Noisy Trapdoor Claw-free Functions *(abbreviated e^{3}NTCF) if*

1. *$\mathcal{F}$ is a TCF family with an AHCB property.*
2. *$\mathcal{G}$ is an injective trapdoor one-way function family.*
3. *$\mathcal{H}$ is an extractable one-way function family, in the sense that for any algorithm that takes the description of $h \in \mathcal{H}$ as input and outputs $y \in \text{range}(h)$, there exists an extractor which outputs a preimage x such that $h(x) = y$.*
4. *The functions are computationally indistinguishable from each other. In other words, given a description of a function from one of the three families, no polynomial-time classical algorithm can determine the function's type with probability non-negligibly greater than $1/3$.*

[8] This also serves to exclude the "extended-domain" attack mentioned in Footnote 4.

[9] As usual, this primitive depends on a security parameter λ, which we suppress for readability.

For the formal statement, see [5, Definition 8].

The notion of an e^{3}NTCF function family is an extension of Injective Invariant Noisy TCFs, which were introduced in [38] to derive a protocol for verifying general quantum computations, and which only use the first two types of families $\mathcal{F}$ and $\mathcal{G}$.

The preceding discussion naturally leads to the single-round Protocol 2 (see [5, Protocol 4] for the formal protocol), based on an e^{3}NTCF. Our main result is that this protocol is a proof of quantumness, i.e. that an efficient quantum prover can succeed with high probability, but no efficient classical prover can.

Algorithm 2 Single-Round Proof of Quantumness based on e^{3}NTCF. (informal)

1. The verifier samples a challenge type $a \leftarrow_U \{\texttt{Eq}, \texttt{sIm}, \texttt{wIm}\}$.
   if $a = \texttt{Eq}$:
2. Verifier samples $f \in \mathcal{F}$ with trapdoor t_f and sends a description of f to the prover.
3. Verifier receives tuple (y, d) and accepts iff $d \cdot (x_0 \oplus x_1) = 0$ (and $d \neq 0$). Here, (x_0, x_1) are the two preimages of y, which the verifier can efficiently compute with t_f.
   else if $a = \texttt{sIm}$:
4. Verifier samples $g \in \mathcal{G}$ with trapdoor t_g and sends a description of g to the prover.
5. Verifier receives tuple (y, d) and accepts iff $y \in g(\mathcal{X})$. This check is efficient using t_g.
   else if $a = \texttt{wIm}$:
6. Verifier samples $h \in \mathcal{H}$ with trapdoor t_h and sends a description of h to the prover.
7. Verifier receives tuple (y, d) and accepts iff $y \in h(\mathcal{X}')$. This check is efficient using t_h.
   end if

▶ **Theorem 2** (informal). *Suppose that $(\mathcal{F}, \mathcal{G}, \mathcal{H})$ is an e^{3}NTCF. Then Protocol 2 is a proof of quantumness, i.e.,*

1. ***Completeness.*** *There exists an efficient quantum prover that succeeds with probability $1 - \text{negl}(\lambda)$.*
2. ***Soundness.*** *Every efficient classical prover succeeds with probability at most $5/6 + \text{negl}(\lambda)$.*

In the theorem, λ is the security parameter. As usual, the families $(\mathcal{F}, \mathcal{G}, \mathcal{H})$ implicitly depend on the security parameter and "efficient" classical or quantum provers are those whose runtime scales polynomially in λ. For the formal statement, see [5, Theorem 4].

Proof Sketch. We have already sketched the security proof when we explained why we need to introduce the three different function families $\mathcal{F}, \mathcal{G}$, and $\mathcal{H}$. We briefly summarize the role that each of these function families and the associated challenge types $\texttt{Eq}, \texttt{sIm}$, and $\texttt{wIm}$ play, and provide a high-level explanation of how the soundness constant $5/6$ is derived.

- The *weak image test* (challenge type $\texttt{wIm}$) uses the extractable function family $\mathcal{H}$. This test ensures that for any classical prover in the protocol, there exists an extractor that extracts a preimage $x \in \mathcal{X}'$ to the output $y = h(x)$ produced by the prover. Note that as discussed above, this preimage might in principle be from the extended domain $\mathcal{X}'$, not just $\mathcal{X}$.
- The *strong image test* (challenge type $\texttt{sIm}$) uses the injective function family $\mathcal{G}$. This test, combined with the indistinguishability of the function families $\mathcal{F}, \mathcal{G}$, and $\mathcal{H}$, ensures that the prover evaluates any function f, g, or h only on inputs in the restricted domain $\mathcal{X}$. Furthermore, from this we can prove that for any $y \in \mathcal{Y}$ that has a preimage in $\mathcal{X}$ under a given *TCF function* f, the extractor will output such a preimage (rather than a different preimage in $\mathcal{X}' \setminus \mathcal{X}$).

- The *equation test* (challenge type `Eq`) uses the TCF function family $\mathcal{F}$ that has the AHCB property. The verifier checks that the prover's output (y, d) satisfies the equation $d \cdot (x_0 \oplus x_1) = 0$, with (x_0, x_1) the preimages of y. From the strong image test, we also know that the extractor for a successful classical prover will produce a preimage $x \in \mathcal{X}$ to y. Together, (x, y, d) would break the AHCB property of f, so no classical prover can win with very high probability.

The soundness factor of 5/6 arises from the fact that the adaptive hardcore bit property only gives an upper bound of 1/2 for the event that the prover fulfills the conditions of all three tests simultaneously, given a function of the family $\mathcal{F}$. Using the computational indistinguishability of the function families $\mathcal{F}, \mathcal{G}, \mathcal{H}$, and the relation

$$\Pr[A] + \Pr[B] \leq 1 + \Pr[A \wedge B]$$

we show that the overall success probability of a classical prover $\mathcal{A}$ in the prover can roughly be bounded as

$$\frac{1}{3}\Big(\Pr_{f\leftarrow\mathcal{F}}\Big[S_{\mathcal{A}}^{\mathtt{Eq}}\Big] + \Pr_{g\leftarrow\mathcal{G}}\Big[S_{\mathcal{A}}^{\mathtt{sIm}}\Big] + \Pr_{h\leftarrow\mathcal{H}}\Big[S_{\mathcal{A}}^{\mathtt{wIm}}\Big]\Big) \leq \frac{1}{3}\Big(2 + \Pr_{f\leftarrow\mathcal{F}}\Big[\text{“}S_{\mathcal{A}}^{\mathtt{wIm}} \wedge S_{\mathcal{A}}^{\mathtt{sIm}} \wedge S_{\mathcal{A}}^{\mathtt{Eq}}\text{”}\Big]\Big) \leq \frac{5}{6}$$

where $S_{\mathcal{A}}^{\mathtt{T}}$ denotes the event that $\mathcal{A}$ passes the test T, and “$S_{\mathcal{A}}^{\mathtt{wIm}} \wedge S_{\mathcal{A}}^{\mathtt{sIm}} \wedge S_{\mathcal{A}}^{\mathtt{Eq}}$” denotes the event that the prover succeeds in the equation test, and is employing a strategy that would succeed in the image tests as well. This corresponds to the event that $\mathcal{A}, \mathcal{A}^*$ produce a valid equation-image pair, which cannot occur with probability higher than 1/2 due to the AHCB property of the function family $\mathcal{F}$. ◀

A key objective in the design of our single-round proofs of quantumness was that the required circuit sizes should not be larger than for multi-round proofs of quantumness. Protocol 2 achieves this: to pass in the protocol, a quantum prover can simply prepare the state $\sum |x\rangle|p(x)\rangle$ for a given function $p \in \mathcal{F} \cup \mathcal{G} \cup \mathcal{H}$, measure the first register in the Hadamard basis to get a string d, and measure the second register in the computational basis to get an image y. These are exactly the same actions as those of an honest prover in the equation test in Protocol 1,[10] except that now the honest prover can perform the entire measurement in one step without experimentally challenging mid-circuit measurements.

2.2 Instantiating e^{3}NTCF families from DDH and LWE

We show that e^{3}NTCF can be instantiated either from DDH and t-KEA, or from LWE and LK-ϵ. Here, we only state the main results and defer a more detailed discussion of the construction to [5, Section 4] and [5, Section 6], respectively.

For the LWE-based construction, we already know that the LWE-based TCF from [13] has an AHCB property. We can combine this with suitable injective and extractable one-way functions. In fact, it turns out that for the LWE-based construction, the roles of the injective and extractable functions in Protocol 2 can be played by the same function family, i.e. we only require two distinct families $(\mathcal{F}, \mathcal{G})$. This simplifies the analysis somewhat, as we explain in [5, Section 5]. Combined with Theorem 2, we obtain the following:

[10] In our constructions of e^{3}NTCF, the functions $g \in \mathcal{G}$ and $h \in \mathcal{H}$ are essentially as costly to evaluate as $f \in \mathcal{F}$ in Protocol 1, so introducing these additional function families does not increase the demands on an honest prover.

▶ **Theorem 3** (Informal). *Assuming the classical intractability of LWE and the lattice knowledge assumption LK-ϵ, there exists a single-round proof of quantumness with completeness* $1 - \mathrm{negl}(\lambda)$ *and soundness* $3/4 + \mathrm{negl}(\lambda)$. *The circuit sizes for an honest prover in this protocol are identical to the circuit sizes in the 2-round protocol from [13].*

While a family of TCFs based on DDH was considered in [29] to construct a 3-round proof of quantumness, it had not been shown that these functions have an AHCB property. We prove this through a lossy sampling technique similar to the proof in [13], showing a reduction from the AHCB property to the *matrix d-linear assumption* [41] (which is implied by standard DDH). We then again augment this TCF family with an injective and an extractable family to get the following result.

▶ **Theorem 4** (Informal). *Assuming the classical intractability of DDH and the knowledge of exponent assumption t-KEA, there exists a single-round proof of quantumness with completeness*[11] $0.99 - \mathrm{negl}(\lambda)$ *and soundness* $5/6 + \mathrm{negl}(\lambda)$. *The circuit sizes for an honest prover in this protocol are identical to the circuit sizes in the 3-round protocol from [29].*

3 Impossibility of black-box reductions

Knowledge assumptions are non-falsifiable cryptographic assumptions [40, 20], which makes their use somewhat undesirable. A natural question is whether our main results, single-round proofs of quantumness, can also be achieved using only standard falsifiable (also called game-based) assumptions such as DDH or LWE, or even weaker assumptions like the existence of one-way functions.

As was observed in [39], the security of a single-round proof of quantumness cannot be reduced to a *quantumly hard* problem like LWE in a black-box manner. The reason for this is simple: if there existed a black-box reduction from a successful classical prover to an algorithm for breaking LWE, we could also apply that reduction to the honest quantum prover in the proof of quantumness. Since the honest quantum prover is required to succeed with high probability, this would give a quantum algorithm for LWE. This rules out the existence of such a black-box reduction.[12]

An extension of this reasoning suggests that the use of knowledge assumptions is still justified even for single-round proofs of quantumness based on computational problems that are quantumly easy: now it is of course not a contradiction to have a quantum algorithm that breaks the underlying cryptographic assumption, but we can argue that implementing an honest quantum prover in such a protocol is no easier than breaking the assumption outright: Suppose we had a single-round proof of quantumness with a black-box reduction to factoring. Then we could again use that reduction with the honest quantum prover to construct a factoring algorithm whose only quantum component is repeatedly (and independently) running the honest prover. In this sense, implementing the honest prover is as hard as breaking factoring: if we had an honest prover that was much easier to implement than Shor's algorithm, we could use that prover and the black-box reduction to construct a much more

[11] The reason completeness here is 0.99 instead of 1 is inherited from [29]. There, the quantum prover does not create a superposition over the exact range of the TCF but over a superset. However, it can be shown that the exact range contains at least a 0.99 fraction of the elements in the superset.

[12] Note that this argument does not apply to interactive protocols, ROM-based protocols, or protocols that use whitebox assumptions like ours: in those cases, one cannot simply run the reduction on the quantum prover as the reduction may perform operations that only work for classical provers, e.g. rewinding or using knowledge extractors.

practical quantum factoring algorithm. In contrast, in the DDH-based proof of quantumness from [29] and also in our single-round version thereof, implementing the honest quantum prover does not require computing discrete logarithms.

4 Related work

As mentioned in Section 1.1, cryptographic proofs of quantumness were introduced in [13]. Since then, there have been a number of follow-up works aiming to understand the types of cryptographic constructions on which these protocols can be based, as well as how to make them more efficient [30, 14, 35, 26, 4, 31, 39]. So far, only small-scale demonstrations of these protocols have been performed [49, 33], though the hope is that by further reducing the sizes of the quantum circuits and the amount of interaction between the verifier and the prover, these protocols can be performed with NISQ devices.

The only existing constructions of single-round proofs of quantumness (apart from trivial ones like asking the prover to factor a large number) use the random oracle model [14, 45]. In both of these prior works, the quantum circuits required to succeed in these protocols do not have to perform mid-circuit measurements. However, instantiating the random oracle with plausible cryptographic hash functions, like SHA-2 or SHA-3, incurs a substantial overhead. The most efficient known quantum implementations of SHA-2 and SHA-3 [28] need a number of qubits on the order of 10^4 and have a quantum gate count on the order of 10^6.

On the other hand, the ROM-based single-round proof of quantumness in [14] only requires a TCF, without the need for the AHCB property. General TCFs have been constructed from a wider variety of cryptographic assumptions, achieving lower asymptotic complexities than known NTCF constructions. Thus, despite the fact that the use of a random oracles introduces a significant overhead, a resource comparison between our LWE-based single-round proof of quantumness and the ROM-based protocol of [14] is not straightforward. A more conclusive answer on which single-round protocol admits the smallest quantum circuit for a demonstration of quantum advantage requires a detailed finite-size analysis, which we leave for future work.

Since our construction of e^3NTCFs makes use of extractable one-way functions (EOWFs), it is natural to ask whether we could have used existing results concerning these functions [16, 8, 10]. In particular, in [8], Bitansky et al. prove the existence of a class of EOWFs from the subexponential hardness of LWE against adversaries with bounded auxilary input, without having to rely on a knowledge assumption. There are at least two obstacles towards applying those results to our setting. First, the extractable one-way functions we require have to be computationally indistinguishable from TCFs. It is unclear whether the EOWFs of [8] can be suitably adapted to satisfy this requirement. Second, since one of the motivations for our work is to devise more efficient single-round proofs of quantumness, we would like to ensure that the honest quantum prover's circuits are at most as large as in the multi-round protocols. However, constructing a family of e^3NTCFs with the EOWFs in [8] would likely require larger circuits.

Knowledge assumptions in the quantum setting have also been considered in [34] to derive *quantum money* and *quantum lightning* schemes. There, the assumptions are required to be sound against quantum adversaries. This introduces some challenges in formalizing the appropriate notion of a quantum knowledge assumption. The subsequent work of [46] also discusses this point. Interestingly, this latter work points out how certain knowledge assumptions can be generically broken by a quantum adversary. In some sense, this is exactly what we exploit to arrive at our single-round proofs of quantumness.

5 Discussion and open problems

We have shown how existing knowledge assumptions, together with standard cryptographic assumptions, lead to efficient single-round proofs of quantumness. Our work opens up several avenues for future research.

One such avenue is the possibility of a *white-box proof* for a single-round proof of quantumness based only on the classical intractability of, say, LWE. This would circumvent the impossibility discussed in Section 3. As mentioned in the previous section, one possible approach would be to use the extractable one-way functions based on subexponential LWE from [8], with the caveat that this would provide soundness against classical provers with bounded auxiliary input. We leave this as an interesting but challenging open problem.

Currently, our construction necessitates the use of a TCF which satisfies the AHCB property. This is a fairly strong requirement which so far has only been achieved from LWE [13], isogeny-based group actions [3], and, in this work, the DDH assumption. It would be desirable to use TCFs that are not known to have an AHCB property, such as those based on Ring-LWE or Rabin's function (x^2 mod N), since those require smaller circuits than the TCFs based on LWE or DDH, as outlined in [29, 30]. However, it is unclear if our construction can be modified so as to not require the AHCB property; in the equation test, the prover is allowed to output *any* valid equation, hence the need for an *adaptive* hardcore bit. The protocol in [29] is able to circumvent this and use TCFs without an AHCB by introducing an additional round of interaction between the verifier and the prover, leading to a 3-round protocol. Unfortunately, there does not seem to be an obvious way of employing knowledge assumptions to reduce the round complexity of the protocol from [29]. Thus, a fundamentally different approach would be needed to achieve a single-round protocol without relying on the AHCB property. Of course, for the purpose of having more practical protocols, proving an AHCB statement for the TCFs based on Ring-LWE or Rabin's function (even relying on knowledge assumptions) would be equally useful.

We remark that the LWE-based approach is expected to be more efficient than the DDH one. This is because, as is also discussed in [29], the DDH approach requires performing a large number of group exponentiations coherently. It would therefore seem that the DDH approach is less efficient than performing Shor's algorithm to solve the discrete logarithm problem. However, as [29] points out, the proof of quantumness protocol can benefit from certain optimizations that are not known to be possible for Shor's algorithm. As an example, the group exponentiation circuits for the DDH-based proof of quantumness need not be reversible, thereby halving the depth and number of gates compared to Shor's algorithm (see section III.D in [29] for more details).

From existing knowledge assumptions one is only able to prove soundness against classical adversaries, i.e. one is able to bound the success probability of a classical prover, but one cannot make a statement about the internal operations of a successful quantum prover. This is sufficient for a proof of quantumness, but other protocols, such as certifiable randomness generation [13], remote state preparation [21], or verification of quantum computations [38], require soundness against quantum adversaries, i.e. one needs to have at least a partial characterization of any quantum adversary in the protocol. As mentioned, one can derive single-round proofs of quantumness in the random oracle model, and these can be extended to single-round versions of quantum-sound protocols by proving security in the quantum random oracle model [11, 14, 2, 47]. However, with knowledge assumptions, it is unclear what the right quantum-sound analogue would be to derive these more advanced functionalities. As mentioned in the previous section, quantum knowledge assumptions have been considered

in [34, 46] to construct quantum money and quantum lightning schemes. These could serve as a useful starting point for constructing single-round quantum protocols for functionalities like single-device randomness expansion. We leave formalizing this for future work.

References

1 Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 333–342, 2011. doi:10.1145/1993636.1993682.

2 Gorjan Alagic, Andrew M Childs, Alex B Grilo, and Shih-Han Hung. Non-interactive classical verification of quantum computation. In *Theory of Cryptography Conference*, pages 153–180. Springer, 2020. doi:10.1007/978-3-030-64381-2_6.

3 Navid Alamati, Giulio Malavolta, and Ahmadreza Rahimi. Candidate trapdoor claw-free functions from group actions with applications to quantum protocols. In *Theory of Cryptography Conference*, pages 266–293. Springer, 2022. doi:10.1007/978-3-031-22318-1_10.

4 Yusuf Alnawakhtha, Atul Mantri, Carl A Miller, and Daochen Wang. Lattice-based quantum advantage from rotated measurements. *arXiv preprint*, 2022. doi:10.48550/arXiv.2210.10143.

5 Petia Arabadjieva, Alexandru Gheorghiu, Victor Gitton, and Tony Metger. Single-round proofs of quantumness from knowledge assumptions, 2024. doi:10.48550/arXiv.2405.15736.

6 Frank Arute et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019. doi:10.1038/s41586-019-1666-5.

7 Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 326–349, New York, NY, USA, 2012. Association for Computing Machinery. doi:10.1145/2090236.2090263.

8 Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. On the existence of extractable one-way functions. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 505–514, 2014. doi:10.1145/2591796.2591859.

9 Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Omer Paneth, and Rafail Ostrovsky. Succinct non-interactive arguments via linear interactive proofs. In *Theory of Cryptography: 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013. Proceedings*, pages 315–333. Springer, 2013. doi:10.1007/s00145-022-09424-4.

10 Nir Bitansky, Noa Eizenstadt, and Omer Paneth. Weakly extractable one-way functions. In *Theory of Cryptography: 18th International Conference, TCC 2020, Durham, NC, USA, November 16–19, 2020, Proceedings, Part I 18*, pages 596–626. Springer, 2020. doi:10.1007/978-3-030-64375-1_21.

11 Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *Advances in Cryptology–ASIACRYPT 2011: 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings 17*, pages 41–69. Springer, 2011. doi:10.1007/978-3-642-25385-0_3.

12 Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. On the complexity and verification of quantum random circuit sampling. *Nature Physics*, 15(2):159–163, 2019. doi:10.1038/s41567-018-0318-2.

13 Zvika Brakerski, Paul F. Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 320–331. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00038.

14 Zvika Brakerski, Venkata Koppula, Umesh Vazirani, and Thomas Vidick. Simpler proofs of quantumness, 2020. doi:10.48550/arXiv.2005.04826.

15 Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of computer and system sciences*, 37(2):156–189, 1988. doi:10.1016/0022-0000(88)90005-0.

16 Ran Canetti and Ronny Ramzi Dakdouk. Towards a theory of extractable functions. In *Theory of Cryptography Conference*, pages 595–613. Springer, 2009. doi:10.1007/978-3-642-00457-5_35.

17 Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *Advances in Cryptology — CRYPTO '91*, pages 445–456, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg. doi:10.1007/3-540-46766-1_36.

18 Thomas Debris-Alazard, Pouria Fallahpour, and Damien Stehlé. Quantum oblivious LWE sampling and insecurity of standard model lattice-based snarks. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 423–434. ACM, 2024. doi:10.1145/3618260.3649766.

19 Rosario Gennaro, Michele Minelli, Anca Nitulescu, and Michele Orrù. Lattice-based zk-SNARKs from square span programs. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, pages 556–573, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3243734.3243845.

20 Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 99–108, 2011. doi:10.1145/1993636.1993651.

21 Alexandru Gheorghiu and Thomas Vidick. Computationally-secure and composable remote state preparation. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1024–1033. IEEE, 2019. doi:10.1109/FOCS.2019.00066.

22 Craig Gidney and Martin Ekerå. How to factor 2048 bit rsa integers in 8 hours using 20 million noisy qubits. *Quantum*, 5:433, 2021. doi:10.22331/q-2021-04-15-433.

23 Oded Goldreich and Johan Håstad. On the complexity of interactive proofs with bounded communication. *Inf. Process. Lett.*, 67(4):205–214, 1998. doi:10.1016/S0020-0190(98)00116-1.

24 Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. COMPUT*, 18(1):186–208, 1989. doi:10.1145/22145.22178.

25 Élie Gouzien and Nicolas Sangouard. Factoring 2048-bit rsa integers in 177 days with 13 436 qubits and a multimode memory. *Physical review letters*, 127(14):140503, 2021. doi:10.1103/PhysRevLett.127.140503.

26 Shuichi Hirahara and François Le Gall. Test of quantumness with small-depth quantum circuits. *arXiv preprint*, 2021. doi:10.48550/arXiv.2105.05500.

27 Yuval Ishai, Hang Su, and David J. Wu. Shorter and faster post-quantum designated-verifier zkSNARKs from lattices. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, pages 212–234, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3460120.3484572.

28 Kyungbae Jang, Sejin Lim, Yujin Oh, Anubhab Baksi, Sumanta Chakraborty, and Hwajeong Seo. Quantum implementation and analysis of sha-2 and sha-3. Cryptology ePrint Archive, Paper 2024/513, 2024. URL: https://eprint.iacr.org/2024/513.

29 Gregory D. Kahanamoku-Meyer, Soonwon Choi, Umesh V. Vazirani, and Norman Y. Yao. Classically verifiable quantum advantage from a computational bell test. *Nature Physics*, 18(8):918–924, August 2022. doi:10.1038/s41567-022-01643-7.

30 Gregory D Kahanamoku-Meyer and Norman Y Yao. Fast quantum integer multiplication with zero ancillas. *arXiv preprint*, 2024. doi:10.48550/arXiv.2403.18006.

31 Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Lisa Yang. Quantum advantage from any non-local game. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1617–1628, 2023. doi:10.1145/3564246.3585164.

32 Thomas Kerber, Aggelos Kiayias, and Markulf Kohlweiss. Composition with knowledge assumptions. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, pages 364–393, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-84259-8_13.

33 Laura Lewis, Daiwei Zhu, Alexandru Gheorghiu, Crystal Noel, Or Katz, Bahaa Harraz, Qingfeng Wang, Andrew Risinger, Lei Feng, Debopriyo Biswas, et al. Experimental implementation of an efficient test of quantumness. *Physical Review A*, 109(1):012610, 2024. doi:10.1103/PhysRevA.109.012610.

34 Jiahui Liu, Hart Montgomery, and Mark Zhandry. Another round of breaking and making quantum money: How to not build it from lattices, and more. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 611–638. Springer, 2023. doi:10.1007/978-3-031-30545-0_21.

35 Zhenning Liu and Alexandru Gheorghiu. Depth-efficient proofs of quantumness. *Quantum*, 6:807, 2022. doi:10.22331/q-2022-09-19-807.

36 Jake Loftus, Alexander May, Nigel P. Smart, and Frederik Vercauteren. On cca-secure somewhat homomorphic encryption. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography*, pages 55–72, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. doi:10.1007/978-3-642-28496-0_4.

37 Lars S Madsen, Fabian Laudenbach, Mohsen Falamarzi Askarani, Fabien Rortais, Trevor Vincent, Jacob FF Bulmer, Filippo M Miatto, Leonhard Neuhaus, Lukas G Helt, Matthew J Collins, et al. Quantum computational advantage with a programmable photonic processor. *Nature*, 606(7912):75–81, 2022. doi:10.1038/s41586-022-04725-x.

38 Urmila Mahadev. Classical verification of quantum computations, 2018. doi:10.48550/arXiv.1804.01082.

39 Tomoyuki Morimae and Takashi Yamakawa. Quantum advantage from one-way functions. *arXiv preprint*, 2023. doi:10.48550/arXiv.2302.04749.

40 Moni Naor. On cryptographic assumptions and challenges. In *Annual International Cryptology Conference*, pages 96–109. Springer, 2003. doi:10.1007/978-3-540-45146-4_6.

41 Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. Cryptology ePrint Archive, Paper 2009/105, 2009. doi:10.1007/978-3-642-03356-8_2.

42 John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018. doi:10.22331/q-2018-08-06-79.

43 Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999. doi:10.1137/S0097539795293172.

44 Yulin Wu, Wan-Su Bao, Sirui Cao, Fusheng Chen, Ming-Cheng Chen, Xiawei Chen, Tung-Hsun Chung, Hui Deng, Yajie Du, Daojin Fan, et al. Strong quantum computational advantage using a superconducting quantum processor. *Physical review letters*, 127(18):180501, 2021. doi:10.1103/PhysRevLett.127.180501.

45 Takashi Yamakawa and Mark Zhandry. Verifiable quantum advantage without structure. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 69–74. IEEE, 2022. doi:10.1109/FOCS54457.2022.00014.

46 Mark Zhandry. Quantum money from abelian group actions. *arXiv preprint*, 2023. doi:10.48550/arXiv.2307.12120.

47 Jiayu Zhang. Classical verification of quantum computations in linear time. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 46–57. IEEE, 2022. doi:10.1109/FOCS54457.2022.00012.

48 Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, et al. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, 2020. doi:10.1126/science.abe8770.

49 Daiwei Zhu, Gregory D Kahanamoku-Meyer, Laura Lewis, Crystal Noel, Or Katz, Bahaa Harraz, Qingfeng Wang, Andrew Risinger, Lei Feng, Debopriyo Biswas, et al. Interactive cryptographic proofs of quantumness using mid-circuit measurements. *Nature Physics*, 19(11):1725–1731, 2023. doi:10.1038/s41567-023-02162-9.

The Local Hamiltonian Problem for Quasi-Quantum States: A Toy Model for the Quantum PCP Conjecture

Itai Arad
Centre for Quantum Technologies, Singapore

Miklos Santha
Centre for Quantum Technologies, Singapore

Abstract

In this work we define a new classical constraint satisfaction problem that shares many of the properties of the quantum local Hamiltonian problem, distinguishing it from the usual classical k-SAT problem. The problem consists of minimizing the number of violated local constraints over a restricted set of *distributions of assignments*. We show that these distributions can be 1-to-1 mapped to a superset of the quantum states, which we call *k-local quasi-quantum states*. Nevertheless, we claim that our optimization problem is essentially classical, by proving that it is an NP-complete problem. Interestingly, the optimal distribution shares many of the properties of quantum states. In particular, it is not determined straightforwardly by its local marginals, and consequently, it can be used as a classical toy model to study several aspects of Hamiltonian complexity that are different from their classical counter parts. These include the complexity of 1D systems (which is in P for classical CSPs, but is QMA-hard for quantum systems), and the lack of an easy search-to-decision reduction. Finally, we believe that our model can be used to gain insights into the quantum PCP conjecture. Indeed, while we have shown that approximating the minimal number of unsatisfiable constraints to within an $\Theta(1)$ is NP-hard, it is not clear if the problem remains hard if we want to approximate the minimal *fraction* of unsatisfiable constraints to within an $\Theta(1)$; as in the quantum PCP conjecture, naive quantization of the classical proofs does not seem to work.

2012 ACM Subject Classification Theory of computation → Complexity classes; Theory of computation → Problems, reductions and completeness; Theory of computation → Proof complexity; Theory of computation → Complexity theory and logic; Theory of computation → Quantum complexity theory

Keywords and phrases Quantum Complexity, Probability Checkable Proofs, Local Hamiltonians

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.9

Category Extended Abstract

Related Version *Full Version*: `https://arxiv.org/abs/2410.13549`

Funding This research project was supported by the National Research Foundation, Singapore and A*STAR under its CQT Bridging Grant.

Acknowledgements We thank Dima Gavinsky, Thomas Vidick, Chen Zhili and Anurag Anshu for valuable discussions.



© Itai Arad and Miklos Santha;
licensed under Creative Commons License CC-BY 4.0



Algorithmic Collusion Without Threats

Eshwar Ram Arunachaleswaran
Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA

Natalie Collina
Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA

Sampath Kannan
Simons Institute for the Theory of Computing, University of California, Berkeley, CA, USA

Aaron Roth
Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA

Juba Ziani
ISyE, Georgia Tech, Atlanta, GA, USA

Abstract

There has been substantial recent concern that automated pricing algorithms might learn to "collude." Supra-competitive prices can emerge as a Nash equilibrium of repeated pricing games, in which sellers play strategies which threaten to punish their competitors if they ever "defect" from a set of supra-competitive prices, and these strategies can be automatically learned. But threats are anti-competitive on their face. In fact, a standard economic intuition is that supra-competitive prices emerge from *either* the use of threats, *or* a failure of one party to correctly optimize their payoff. Is this intuition correct? Would explicitly preventing threats in algorithmic decision-making prevent supra-competitive prices when sellers are optimizing for their own revenue?

No. We show that supra-competitive prices can robustly emerge even when both players are using algorithms which do not explicitly encode threats, and which optimize for their own revenue. Since deploying an algorithm is a form of commitment, we study sequential Bertrand pricing games (and a continuous variant) in which a first mover deploys an algorithm and then a second mover optimizes within the resulting environment. We show that if the first mover deploys *any* algorithm with a no-regret guarantee, and then the second mover even approximately optimizes within this now static environment, monopoly-like prices arise. The result holds for any no-regret learning algorithm deployed by the first mover and for *any* pricing policy of the second mover that obtains them profit at least as high as a random pricing would – and hence the result applies even when the second mover is optimizing only within a space of non-responsive pricing distributions which are incapable of encoding threats. In fact, there exists a set of strategies, neither of which explicitly encode threats that form a Nash equilibrium of the simultaneous pricing game in algorithm space, and lead to near monopoly prices. This suggests that the definition of "algorithmic collusion" may need to be expanded, to include strategies without explicitly encoded threats.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory

Keywords and phrases Algorithmic Game Theory, Algorithmic Collusion, No-Regret Dynamics

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.10

Related Version *Full Version*: `https://www.arxiv.org/pdf/2409.03956` [3]

Supplementary Material *Software (Source Code)*: `https://github.com/eshwarram/Non_Myopic_Pricing`, archived at `swh:1:dir:8ecaca48118cc1984e5a41878d828d807920a98b`

Funding ERA was supported by NSF grants CCF 1910534 and 2045128. NC and AR were partially supported by NSF grants FAI-2147212 and CCF-2217058, the Hans Sigrist Prize, and the Simons Collaboration on Algorithmic Fairness.



© Eshwar Ram Arunachaleswaran, Natalie Collina, Sampath Kannan, Aaron Roth, and Juba Ziani; licensed under Creative Commons License CC-BY 4.0



Acknowledgements We thank Rakesh Vohra and Deke Hill for valuable discussions on the subject of algorithmic collusion. SK was on leave from the University of Pennsylvania and serving as the Associate Director of the Simons Institute for the Theory of Computing at the time of writing of this paper. Finally, we would like to thank the ITCS reviewers for their helpful comments and feedback.

1 Introduction

Consider a market in which there are two sellers of some commodity. We hope and expect that competition should drive the price of the good in the market down to its production cost – for if seller A was consistently selling above cost, then there would be an opportunity for seller B to slightly undercut seller A, capturing the whole market and increasing their profit. In fact, this is the Nash (and correlated) equilibrium outcome in several classical formalizations of such competition including the Bertrand Duopoly model. We refer to pricings that are at (or near) the Nash equilibrium prices as *competitive prices*.

As the name suggests, competition is essential to maintaining competitive prices: a monopolist could freely set the price of a good as high as the market could support (which, for goods with inelastic demand, could be quite high indeed) without fear of being undercut by a competitor. So, if in a market with multiple participants, we see prices that are closer to the monopoly price than the competitive price, we might suspect "anti-competitive" behavior on the part of the sellers.

Yet, defining anti-competitive behavior turns out to be a difficult exercise. The Sherman act has been interpreted to require express agreement to coordinate on prices in the form of overt communication as a pre-requisite for liability. However, a recent line of work has focused on the ability of *pricing algorithms* to arrive at supra-competitive prices without explicit communication – concerns rooted in real world observations [4] and a growing academic literature [2, 20, 15]. The mechanism underlying most results in this literature ultimately are rooted in "folk-theorem" style equilibria of repeated pricing games [6, 22]. Broadly speaking, folk theorems establish that all pairs of prices that guarantee the sellers at least the profit they obtain with competitive pricing can be realized as equilibria of the repeated game. Crucially, the strategies which are shown to lead to these anti-competitive equilibria have explicitly encoded threats aimed at punishing the other seller if they deviate from a proscribed pricing path. Further, reinforcement learning algorithms can automatically discover these strategies; recent work [8, 18] has shown both empirically and analytically that these algorithms can converge to supra-competitive prices, and they do so by the emergence of punishments and threats, even without being explicitly programmed to learn such strategies. Using such algorithms may not violate current anti-trust law as they do not involve explicit communication, but one can imagine future legislation and case law that forbids the use of algorithms that make use of threats (appropriately defined). An algorithm is after all an explicit commitment to action as a function of its observations, and committing to an algorithm might be interpreted as communicating the content of the threat encoded within the algorithm.

To formally rule out "anti-competitive" behavior, an approach one might take is to define algorithms that do not encode threats. In fact, the standard economic stance [16, 8] has been that threats are an *explicit* requirement for collusion, distinguished from other situations in which supra-competitive prices can arise due to a "failure to optimize:"

> "To us economists, collusion is not simply a synonym of high prices but crucially involves "a reward-punishment scheme designed to provide the incentives for firms to consistently price above the competitive level" (Harrington 2018, p. 336). The reward-punishment scheme ensures that the supra-competitive outcomes may be obtained in equilibrium and do not result from a failure to optimize." [8]

A first attempt might be to say that (at least if there are other optimizers in the market) "non-responsive" algorithms which set prices completely independently of competitor prices (and consequences of competitor prices) should be permitted – indeed, if they cannot react to their competitors, then they cannot threaten them. But this is far too limiting as it does not allow for algorithms that react to market conditions. Recent work by [17] and [10] set out a principled definition of what constitutes non-collusive behavior of a pricing algorithm by defining algorithms that satisfy some internal consistency properties (calibrated regret, also known as "swap regret") and then shows that if all algorithms in the market satisfy these conditions then the outcome will be competitive prices. Can this condition be relaxed? What if the first entrant to a market deploys (and hence commits to) an algorithm that satisfies these internal consistency conditions, and then the second entrant plays a simple policy that does not satisfy the swap regret condition, but also does not encode threats (e.g. because it is entirely non responsive)? Might we still see competitive prices? Would we, in accordance with the intuition of [16, 8] see competitive prices if the follower not only did not deploy threats, but also succeeded in "optimizing"?

In this paper we show that the answer is robustly "no" by showing how a broad class of "reasonable" behavior on the part of all parties must lead to supra-competitive (near monopoly) prices. In particular, with the view that deploying an algorithm is inherently a "commitment" (at least in the short term), we study two seller pricing games in a leader-follower model, in which the leader first deploys an algorithm, and then the follower deploys an algorithm which may optimize within the market environment that is implicitly defined by the leader's pricing algorithm. If both sellers viewed this scenario as a Stackelberg game, and optimized over the set of all pricing algorithms, it would hardly be surprising if anti-competitive behavior emerged. Indeed, the Stackelberg equilibria in algorithm space (See [11]) in a repeated pricing game would involve the leader committing to an algorithm that prescribes monopoly prices and commits to punish the follower if she deviates from this prescription.

However, we instead study the behavior that emerges if sellers attempt to play reasonable algorithms. In particular, we assume the leader (following the principle set out by [17, 10]) commits to setting prices using an arbitrary *no-regret* algorithm. No-Regret algorithms are widely understood to be a "natural" class of algorithms for use in competitive environments [24], and our results will hold for any algorithm in this class, which includes the no-swap regret algorithms studied by [17, 10].

Once the leader deploys a no-regret learning algorithm, the other seller (whom we call the follower, or the optimizer) deploys an algorithm to optimize their own revenue in the environment defined by the leader's no-regret learning algorithm. Once again, we do not assume that the follower is best responding in algorithm space: they may, e.g., use a heuristic reinforcement learning algorithm to optimize over a limited set of policies – even *non-responsive* policies that are explicitly independent of the leader's prices, and hence cannot encode threats. We show that however they choose to optimize their own revenue, if they are even marginally successful (choosing any policy that guarantees them at least the revenue that a random pricing strategy would), then in combination, the two algorithms will inevitably lead to supra-competitive prices in a variety of settings. When this happens, it is not just the follower who enjoys the revenue of the high prices, but as we prove, also the leader (if the leader deployed a no-regret learning algorithm). Thus the leader will have no strong incentive to deviate from their deployed algorithm, even if they deployed it without initially being aware of the competitive environment. In fact, within the class of "reasonable" strategies we study for the leader and follower (a no-swap-regret algorithm for the leader

and a non-responsive algorithm for the follower), there exists a Nash equilibrium of the simultaneous pricing game in algorithm space, which gives *no party* any incentive to deviate, supports near-monopoly prices, and does not involve threats. This suggests that it may be worth revisiting the standard economic viewpoint [8, 16] that *threats* are a necessary precondition for collusion.

Summary of Results

We study two types of models: a simple Bertrand duopoly model, where the firm with the lowest price captures the entire demand, and a smoother multinomial-logit-based model where firms with higher prices can still capture some of the demand. Our multinomial logit model includes a temperature parameter τ that controls how sensitive the demand for each firm is to prices, and converges to a Bertrand competition when the parameter τ grows large. Within these models, we prove the following results:

1. The second entrant into the market (whom we call the optimizer) can extract a constant fraction of the monopoly price revenue against any no-regret algorithm by playing a non-responsive strategy (i.e. a fixed distribution of prices across all rounds, that does not depend on or respond to the history of play). As a result, if they approximately optimize their pricing policy (in any set of policies that includes natural fixed price distributions), they will be guaranteed a constant fraction of the monopoly revenue. Observe that this holds even if the class they optimize over *only* includes non-responsive pricing strategies, which are incapable of encoding threats.
2. Simultaneously, by deploying any no-regret algorithm, the first entrant (whom we call the learner) extracts a constant fraction of the revenue obtained by any optimizer strategy that obtains a constant fraction of the monopoly price revenue. Taken in conjunction with the previous result, this implies that if the first entrant into the market deploys (and hence commits to) a no-regret learning algorithm, and the optimizer deploys any strategy that approximately optimizes over any class of policies that includes fixed price distributions, *both* parties will extract a constant fraction of the monopoly price revenue. This has important implications: not only do supra-competitive prices emerge, but the benefits are shared across both sellers, giving neither a substantial incentive to deviate. In other words, the optimizer is not "exploiting" the no-regret learner, but rather implicitly cooperating with them, despite the absence of threats.
3. In fact, we show that if the first entrant into the market deploys a no-swap regret algorithm, then there is a non-responsive (i.e. fixed) price distribution for the optimizer that forms a *Nash* equilibrium of the simultaneous pricing game in algorithm space: i.e. *neither* player has any incentive to deviate from their selected algorithm. This demonstrates the existence of a Nash equilibrium of the game that supports supra-competitive prices and yet does not involve threats, contra to the economic intuition [8, 16] that supra-competitive prices result from *either* threats or a failure to optimize.

When we refer to supra-competitive prices in the above theoretical results, we mean prices that are within a constant factor of the monopoly price. The constants in our theorems are small, but we provide numerical evidence that the constant is in fact no smaller than $2/e \approx 0.74$.

We also prove an additional result about the Bertrand duopoly model:

- Our results apply to *any* no regret learning algorithm deployed by the first entrant into the market. This is a super-set of the no-swap regret learning algorithms proposed to be definitionally "competitive" by [17, 10]. Should no regret algorithms beyond no-swap

regret algorithms also be considered competitive? This is not the case for all no-regret dynamics – [23] show the existence of coarse correlated equilibria supported on supra-competitive prices in Bertrand pricing games, implying the existence of no-regret dynamics that result in such prices. However, we show that when both sellers use Mean-based No-Regret (MBNR) algorithms (a class which includes common no-regret learning algorithms such as multiplicative weights and follow the perturbed leader) to set prices, the empirical distribution of play always converges to the competitive price. This suggests that at least in the Bertrand Duopoly model, mean based no-regret learning algorithms might also be considered reasonable/competitive pricing strategies, in addition to the no-swap regret learners proposed by [17, 10]. We note that this result does not necessarily carry over to the more complex pricing models studied by [17].

Due to space constraints, we defer some proofs to the full version of the paper [3].

1.1 Related Work

The seminal work by [8] experimentally shows that Q-learning algorithms, a simple and central type of reinforcement learning algorithm, can learn to collude with each other via threats in a repeated, simultaneous-action pricing game. These algorithms learn to deviate to lower prices in response to another algorithm doing so, and slowly rise back to higher prices. [18] attain similar results in the sequential repeated pricing game. The authors in these works are specifically concerned with the ability of the algorithms to learn clear threats, as opposed to simply reaching supra-competitive prices. This is because, from the perspective of many economists, such threats are a requirement for collusion. The prevailing view, as espoused by [16] and [8], is that any algorithms which reach supra-competitive prices without threats must be suboptimal, and thus uninteresting as candidate pricing algorithms.

This perspective is supported by [1] (experimentally) and [2] (theoretically). These works consider a different class of reinforcement learning algorithms and show that, while they do not converge to supra-competitive prices when they have full computational capacity, they can reach supra-competitive prices if these algorithms are limited in their ability to explore. These supra-competitive prices arise even though these algorithms do not retain enough state information to encode threats. Therefore, while these works do show supra-competitive prices without threats, they are only shown arising from algorithms which are specifically designed to be suboptimal.

Another line of research has explored the emergence of supra-competitive prices as a byproduct of algorithmic symmetry. [15] show that, if two identical, deterministic algorithms are played against each other, their "exploration" will be symmetric, and hence they will be unable to ever observe the benefits of undercutting their competitors (an inherently asymmetric outcome).

[5] study algorithms that attempt to estimate the payoff of actions through stochastic experiments. They show that in certain settings in which players update the reward for the action they play, but not other actions, then the experiments can correlate the actions of the players, leading to supra-competitive prices. In contrast, the algorithms we study operate in the full information setting (because agents can observe the price played by the other agent, they can estimate the profit they would have made had they played a different price at a given time period), and are designed for adversarial settings, which are well specified in competitive environments: they are designed with the understanding that there are no reward distributions to estimate because profits depend on the evolving actions of an opponent.

In contrast to these previous works, which argue variously that the property of algorithms which leads to supra-competitive pricing are threats and sub-optimality, we argue that the core is in the fact that deploying an algorithm is a form of commitment. This is most in line

with the perspective of [20]. This work treats algorithms as a form of commitment which contributes to collusion. However, the algorithms explored in that work are constrained to simple, two-state automata, which severely limits the strategy space. By contrast, we consider the space of all possible algorithms mapping histories to future prices, and show results for broad families of "reasonable" algorithms – not just equilibrium strategies.

There is also a recent line of work on regulating pricing algorithms. Both [10] and [17] suggest that observed *swap regret* might be taken as a proxy for collusive behavior. They show, in a related but more general model than our own, that if both sellers in a duopoly have vanishing swap regret, the time-average price must converge to the competitive price. Our work highlights the importance of *both* sellers having no-swap regret; as we show, if the first entrant into a market deploys a no swap regret algorithm, then many natural behaviors for the second entrant lead to supra-competitive prices.

Finally, there is a recent line of experimental work demonstrating anti-competitive behavior in large language models, including market division in the Cournot competition model [21] and, more similarly to this work, price fixing in the Bertrand competition model [14]. These works provide evidence for the emergence of algorithmic collusion in the increasingly ubiquitous realm of LLMs, underscoring the importance of understanding the foundations of algorithmic collusion.

The emergence of anti-competitive behavior in no-regret dynamics is also explored in [19], though in a somewhat different setting – this work considers agents deploying no-regret algorithms to bid for them in an auction, and shows that agents misreporting their true item value to the algorithms can lead to collusive outcomes.

2 Model

2.1 The Single-Stage Pricing Game

We will first formally define the payoff model of the stage game, in which both sellers select a price only once and specify the payoff they obtain.

We work in a duopoly model, where two sellers are competing to sell to a single buyer. Both sellers pick from a discrete set of prices $\mathcal{P} = \{\frac{1}{k}, \frac{2}{k}, \cdots 1\}$. Here, k is a discretization parameter, and we assume that $k \geq 20$.

Game definition

We consider two sellers, who we will refer to as the Sellers 1 and 2. The action space of each seller is the set of possible prices in $(0, 1]$, up to some discretization parameter $1/k$. Both sellers can choose to distributions of prices from the set $\mathcal{P}$, and we denote the set of all distributions by $\Delta^{\mathcal{P}}$.

To define our game, we first need to define the concept of an allocation rule. An *allocation rule* is a (potentially randomized) rule that divides demand across the two sellers as a function of their chosen prices:

▶ **Definition 1** (Allocation Rule $C_i(p_1, p_2)$)**.** *An* allocation rule $C_i : \mathcal{P} \times \mathcal{P} \mapsto [0, 1]$ *is a mapping from a price pair* (p_1, p_2) *from the two sellers, respectively, to a fraction of the demand that will be allocated to seller* $i \in \{1, 2\}$*.*

Given an allocation rule that defines and controls market outcomes as a function of prices, we can now define the payoff of a seller in our single-stage game as a function of said allocation rule C.

▶ **Definition 2** (Seller Payoff in the Stage Game). *Given an allocation rule $C(.,.)$, the payoff of seller i is given by*

$$u_i(p_1, p_2) = p_i \cdot C_i(p_1, p_2). \tag{1}$$

When the sellers play distributions, the payoff of the seller is understood to be the expected payoff.

We define the buyer price as the price charged by the seller that is the output of the allocation rule.

▶ **Definition 3** (Average Buyer Price in the Stage Game). *For an allocation rule C and (mixed) strategies p_1, p_2, the average price paid by the buyer is $\mathbb{E}[p_1 C_1(p_1, p_2) + p_2 C_2(p_1, p_2)]$. Note that this means that the average buyer price is equal to the (expected) sum of payoffs of the two sellers.*

Throughout this paper we assume that the *total* demand (added across both sellers) for the item being sold is fixed and independent of the prices. Without loss of generality, this total demand is set to be normalized to 1. As a result, the "monopoly price" in these models is always 1. Within this fixed-demand model, we consider two allocation rules: a Bertrand model and a multinomial-logit-based model, both formalized in Section 2.1.1. Both allocation rules are *symmetric* and therefore the resulting games are *symmetric* as well.

We can now formally define the one-stage game studied in this paper, often informally referred to simply as the "stage game":

▶ **Definition 4** (Pricing Stage Game). *A two-player pricing stage game $G(k, C)$ is defined as a combination of:*

1. *Two sellers, respectively named 1, 2;*
2. *An action space $A = \mathcal{P} = \{\frac{1}{k}, \frac{2}{k}, \ldots, 1\}$ corresponding to possible prices that can be played by 1 and 2. We denote by p_1 the action of Seller 1 and p_2 the action of seller 2. These actions may be picked from the set of distributions over $\mathcal{P}$, which we call $\Delta^{\mathcal{P}}$.*
3. *Payoff functions $u_i(p_1, p_2)$ for $i \in \{1, 2\}$, defined as a function of the competitive allocation rule C as in Definition 2.*

A particular pricing game is defined by the allocation rule C and the discretization parameter k. We will work with symmetric allocation rules (the allocation does not depend on the labels of the sellers), which will naturally induce a symmetric finite game between the two sellers.

The one-stage game described is this section is the starting point of our study, one that helps us define competitive prices as the Nash equilibria outcome of the game. However, the primary object of our study is *repeated* interactions between sellers, described in Section 2.2.

Equilibria of the Stage Game

We will refer to two kinds of equilibria of the stage game which differ in their assumed timing. The *Nash* equilibria characterize outcomes in simultaneous play, whereas *Stackelberg* equilibria characterize outcomes of optimal play when one player may first commit to a strategy, and the other player then responds.

▶ **Definition 5** (Nash Equilibrium of the Stage Game). *A pair of (independent) distributions (p_1, p_2) for the two sellers is said to be an ε-Nash equilibrium if $u_1(p_1, p_2) \geq \max_{p \in \mathcal{P}} u_l(p, p_2) - \varepsilon$ and $u_2(p_1, p_2) \geq \max_{p \in \mathcal{P}} u_2(p_1, p) - \varepsilon$. When $\varepsilon = 0$, this pair is a Nash equilibrium.*

▶ **Definition 6** (Stackelberg Equilibrium of the Stage Game). *A pair of (independent) distributions p_1, p_2 for the two sellers is said to be an ε-Stackelberg equilibrium of the stage game with seller 1 as the leader if*

$$u_2(p_1, p_2) \geq \max_{p \in \mathcal{P}} u_2(p_1, p),$$

and

$$u_1(p_1, p_2) \geq \max_{(q_1, q_2) \in \Delta^{\mathcal{P}}} u_1(q_1, q_2) - \varepsilon$$
$$s.t. \quad u_2(q_1, q_2) \geq \max_{q \in \mathcal{P}} u_2(q_1, q).$$

When $\varepsilon = 0$, this pair is a Stackelberg equilibrium.

2.1.1 Competition Models

Now that we have defined the pricing stage game in full generality, we will instantiate it with two different specific competition rules, which result in two well-known economic pricing models.

2.1.1.1 Bertrand Competition

In the Bertrand Competition model, the seller picking the lower price captures all of the demand. The seller picking the higher price earns nothing. For cases in which sellers chose the same price p_L, they split the market equally.

▶ **Definition 7** (Bertrand Competitive Allocation Rule C_i^B). *C_i^B is a competition rule such that, given player $j \neq i$,*

$$C_i^B(p_1, p_2) = \begin{cases} 1 & p_i < p_j, \\ \frac{1}{2} & p_i = p_j, \\ 0 & p_i > p_j \end{cases} \tag{2}$$

▶ **Definition 8** (Bertrand Pricing Stage Game $G^B(k)$). *The* Bertrand Pricing Game *is a pricing game instantiated with the competition rule C^B and with a pricing discretization of $\frac{1}{k}$.*

The Bertrand model is discontinuous in that the seller who selects the lower price captures the entire market. We also study a generalization of Bertrand competition in which the fraction of the market captured by a seller is a smooth function (a logit function) of the difference in prices set by the two sellers. This model has a temperature parameter τ, and as τ tends to infinity, it recovers the Bertrand competition model as a special case.

2.1.1.2 Multinomial-logit-based Price Competition

This competition model is parameterized by a temperature parameter $\tau \geq 0$

▶ **Definition 9** (Logit Competitive Allocation Rule $C_i^{L,\tau}$). *$C_i^{L,\tau}$ is a competition rule such that for $j \neq i$:*

$$C_i^{L,\tau}(p_1, p_2) = \frac{e^{\tau p_j}}{e^{\tau p_i} + e^{\tau p_j}} \tag{3}$$

▶ **Definition 10** (Multinomial-logit-based Pricing Stage Game $G^B(k, \tau)$)**.** *The* Multinomial-logit-based Pricing Game *is a pricing game instantiated with the competition rule* $C^{L,\tau}$ *and with a pricing discretization of* $\frac{1}{k}$*.*

We prove similar results as in the Bertrand model in this logit model, assuming that the temperature parameter is sufficiently large, i.e., the allocation rule is sufficiently sensitive to differences in price. As $\tau \to \infty$, we recover the Bertrand model in the limit, since the lower price seller wins the demand with probability 1. Likewise, when $\tau = 0$, the sellers each win with probability $1/2$ regardless of their price, so it is to be expected that we need τ to be reasonably large to prove any meaningful results.

This model is a special case of the logit model of [8] and is obtained by setting all "product indices" in their model to be equal, and by removing the outside good from consideration (by setting its product index a_0 to $-\infty$), the latter action fixing the total demand.

For simplicity, in the main body of the paper we prove our results in the Bertrand game (Section 4). We show that they generalize to the Multinomial-Logit-Based pricing game in the full version of the paper.

2.2 The Repeated Pricing Game

We now move from the stage game to the repeated pricing interaction that will be our main setting of interest. The pricing game is played repeatedly over T rounds. During each round t, the two sellers simultaneously pick mixed strategies $p_{1,t},\ p_{2,t}$ over the set of prices $\mathcal{P}$ (these are independent distributions over $\mathcal{P}$). The sellers receive the corresponding utilities equal to the expected payoff of pricing $p_{1,t}$ against $p_{2,t}$ in the stage game. Both sellers observe the full mixed strategy of their opponent. Here, the pricing behavior of each player will be defined by algorithms which map history to the pricing distribution at the next round, and we study this larger interaction as a game in which the strategy space for each player is the algorithm that they will deploy. We formalize this setting below by defining the strategy space and the utility functions.

Algorithms as Strategies

The sellers' strategy space is the space of all algorithms, mapping histories to price distributions at the following round in arbitrary ways.

▶ **Definition 11.** *An algorithm* $\mathcal{A}$ *for the* T *round repeated game is a mapping from the history of play (without loss of generality of only the other seller) to the next round's action, denoted by a collection of* T *functions* $A_1, A_2 \cdots A_T$*, each of which deterministically map the transcript of play (up to the corresponding round) to a mixed strategy to be used in the next round, i.e.,* $A_{1,t}(p_{2,1}, p_{2,2}, \cdots, p_{2,t-1}) = p_{1,t}$ *for the algorithm* $\mathcal{A}_1$ *used by seller 1.*

We define a restricted subclass of algorithms below – algorithms that do not use nor respond to the history of play. Such algorithms, in particular, cannot make use of or respond to threats as they are oblivious to the behavior of their opponent.

▶ **Definition 12.** *If* A_t *is independent of its input for all* t*, we call it* non-responsive*. If in addition* $\mathcal{A}_t$ *outputs the* same *distribution for all* t *we call it static.*

If the two sellers pick algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively, this induces a transcript of T price distribution pairs $((p_{1,1}, p_{2,1}), \cdots (p_{1,t}, p_{2,t}) \cdots (p_{1,T}, p_{2,T}))$ where $A_{1,t}(p_{2,1}, p_{2,2}, \cdots, p_{2,t-1}) = p_{1,t}$ and, similarly, $A_{2,t}(p_{1,1}, p_{1,2}, \cdots, p_{1,t-1}) = p_{2,t}$ for each round t.

Game definition

We again consider two sellers, who we will now refer to as the *learner* and the *optimizer*. The action space of each seller is now the set of possible *algorithms* mapping the history of play to a pricing distribution at the current round.

▶ **Definition 13** (Seller Payoff in the Repeated Game). *Let $p_l^{1:T}$ and $p_o^{1:T}$ represent the sequence of T distributions over prices by the learner and the optimizer, respectively, given $\mathcal{A}_l$ and $\mathcal{A}_o$. Then, the payoff of seller $i \in \{l, o\}$ is*

$$U_i(\mathcal{A}_l, \mathcal{A}_o) = \sum_{t=1}^{T} u_i(p_l^t, p_o^t) \tag{4}$$

We can now formally define the central game studied in this paper:

▶ **Definition 14** (Repeated Pricing Game). *A two-player repeated pricing game $G(k, C, T)$ is defined as a combination of:*

1. *Two players, named the "learner" and the "optimizer";*
2. *A time horizon T specifying the length of the repeated game;*
3. *An action space $A = \mathcal{A}^T$ corresponding to possible sequential pricing algorithms for games of length T;*
4. *Payoff functions $U_l(\mathcal{A}_l, \mathcal{A}_o)$ for the learner and $U_o(\mathcal{A}_l, \mathcal{A}_o)$ for the optimizer, defined as in Definition 13.*

A particular repeated pricing game is defined by the allocation rule C and the discretization parameter k.

Equilibrium Notions in Algorithm Space

Once again, we can study two kinds of equilibria of the repeated game in "algorithm space", depending on the timing of the game: Nash equilibria if play is simultaneous, and Stackelberg equilibria if the learner has commitment power and moves first.

▶ **Definition 15** (Nash Equilibrium in Algorithm Space). *A pair $(\mathcal{A}_l, \mathcal{A}_o)$ of algorithms for the two sellers is said to be an ε-Nash equilibrium in "algorithm space" if $U_l(\mathcal{A}_l, \mathcal{A}_o) \geq \max_{\mathcal{A} \in \mathcal{A}^T} U_l(\mathcal{A}, \mathcal{A}_o) - \varepsilon$ and $U_o(\mathcal{A}_l, \mathcal{A}_o) \geq \max_{\mathcal{A} \in \mathcal{A}^T} U_o(\mathcal{A}_l, \mathcal{A}) - \varepsilon$. When $\varepsilon = 0$, we call this pair a Nash equilibrium in algorithm space.*

▶ **Definition 16** (Stackelberg Equilibria in Algorithm Space). *A pair $(\mathcal{A}_l, \mathcal{A}_o)$ of algorithms for the two sellers is said to be an ε-approximate Stackelberg equilibrium in "algorithm space" if*

$$U_o(\mathcal{A}_l, \mathcal{A}_o) \geq \max_{\mathcal{A} \in \mathcal{A}^T} U_o(\mathcal{A}_l, \mathcal{A}),$$

and

$$U_l(\mathcal{A}_l, \mathcal{A}_o) \geq \max_{(\mathcal{B}_l, \mathcal{B}_o) \in \mathcal{A}^T} U_l(\mathcal{B}_l, \mathcal{B}_o) - \varepsilon$$
$$s.t. \;\; U_o(\mathcal{B}_l, \mathcal{B}_o) \geq \max_{\mathcal{B} \in \mathcal{A}^T} U_o(\mathcal{B}_l, \mathcal{B})$$

When $\varepsilon = 0$, we call this pair a Stackelberg equilibrium in algorithm space.

We investigate a setup in which one seller, who we call the Learner, commits to an algorithm $\mathcal{A}_L$, and then the other seller, who we call the Optimizer, responds with their own algorithm $\mathcal{A}_O$ [1]. This is reminiscent of a Stackelberg game in algorithm space; however, in contrast to standard Stackelberg games, we do not necessarily assume that the Optimizer exactly best-responds to the Learner's algorithm, or that the Learner necessarily deploys the algorithm representing the optimal commitment strategy. Instead, we examine which pricing outcomes arise under various Optimizers of different capacities, and show that, when the Optimizer is responding to a no-regret algorithm (defined in Section 3), a robust set of responses, including exactly optimal but also including approximately optimal non-responsive strategies, lead to supra-competitive prices.

Competitive vs supra-competitive prices

In the economics literature, competitive prices are defined as prices that can emerge from the static Nash equilibria of the pricing game (i.e. Nash equilibria of the stage game, where prices, rather than algorithms, form the strategy space). We formalize this below:

▶ **Definition 17** (Competitive and Supra-competitive prices). *For some pricing model G, let p be the highest price that is in the support of any Nash equilibrium of the stage pricing game. Then, p' is a competitive price if $p' \leq p$. Furthermore, p' is a supra-competitive price if $p' = \Omega_k(1)$* [2].

Supra-competitive prices are sometimes defined as any prices that are strictly above the competitive price. In this work, we use and guarantee a stronger definition, and require supra-competitive prices to be a constant fraction of the maximum possible price (1), even when the competitive price tends to 0 as $k \to \infty$. Thus our definition of supra-competitive prices really refers to "near monopoly" prices.

Throughout the paper we will regularly use high seller payoff as proof of supra-competitive prices. We make that connection explicit here, showing that in both the Bertrand and Logit models, the average price over time is exactly equal to the sum of the average utilities of the two sellers:

▶ **Definition 18** (Average Buyer Price in the Repeated pricing Game). *For an allocation rule C and seller algorithms $\mathcal{A}_l$, $\mathcal{A}_o$, the average buyer price over the rounds is simply the time average of the buyer prices over each round induced by the seller's prices and the allocation rule in the resulting transcript of play (see Definition 3).*

We make a simple observation connecting the average payoffs of the sellers with the average prices over the rounds.

▶ **Observation 19.** *The average buyer price in a repeated pricing game exactly equals the sum of the average payoffs of the sellers.*

[1] We follow the convention in prior work in renaming the leader and follower as the learner and optimizer where one of the key cases considered is the leader playing a no-(swap-)regret algorithm. This is helpful in clarifying the roles of the players, as we will use the fact that the optimizer, playing against a no-swap-regret algorithm, can get their Stackelberg *leader* value in the stage game

[2] The $\Omega_k(1)$ notation means that this quantity remains a constant between 0 and 1 regardless of the growth of k, in contrast to an expression such as $\frac{1}{k}$.

3 Online Learning Preliminaries

Classes of Algorithms

We are interested in three specific classes of algorithms that the learner might commit to: *no-regret algorithms*, *no-swap-regret algorithms*, and *mean-based algorithms*. The learner and optimizer repeatedly play a single-stage game G for T rounds. We refer to the learner's payoff function (in the stage game) as u_L and the actions chosen by the learner and optimizer in round t as $x_t \in \Delta^n$ and $y_t \in \Delta^m$ respectively (the action spaces are distributions over finite sets of actions).While we describe these algorithms from the perspective of the learner, they are also valid algorithms for the optimizer and afford them the same properties against algorithms picked by the learner.

No-regret definitions

We consider a setting in which a learner faces an adversary, here called an "optimizer" that can choose an arbitrary sequence of actions, with knowledge of the learner's algorithm. The learner obtains a payoff $u_L(x_t, y_t)$ as a function of his own action x_t and the optimizer's action y_t. A desirable property is that in hindsight, after the sequence of learner and optimizer actions have been realized, the learner does not *regret* not having played a simple strategy (like consistently playing whichever turned out to be the best fixed action in hindsight). We say that an algorithm $\mathcal{A}$ has *regret* $r(T)$ on some sequence of length T if, regardless of the sequence of optimizer actions, the learner is guaranteed that in hindsight, they obtained payoff at least as high as they could have with the best fixed action, minus $r(T)$.

▶ **Definition 20** ($r(T)$-Regret Algorithm)**.** *An algorithm has worst-case regret $r(T)$ if, for any sequence of actions $(y_1, y_2, \ldots, y_T)$ taken by the optimizer, the total payoff of the learner can be lower bounded by*

$$\sum_{t=1}^{T} u_L(x_t, y_t) \geq \left(\max_{x^* \in [n]} \sum_{t=1}^{T} u_L(x^*, y_t) \right) - r(T).$$

An algorithm is a ***no-regret algorithm*** *if it is an $r(T)$-regret algorithm with $r(T) = o(T)$.*

We also consider a strict subset of the class of no-regret algorithms, the class of *no-swap regret* algorithms. An algorithm $\mathcal{A}$ is a no-swap-regret algorithm if it has the no regret property not just marginally, but *conditionally* on each action it played. This can be formalized by requiring that the learner's cumulative payoff is at least as high as it would have been had they, retrospectively, been able to apply some *swap function* $\pi : [n] \to [n]$ to their sequence of actions in hindsight.

▶ **Definition 21** ($r(T)$-Swap Regret Algorithm)**.** *An algorithm has worst-case swap regret $r(T)$ if, for any sequence of actions $(y_1, y_2, \ldots, y_T)$ taken by the optimizer, the total payoff of the learner can be lower bounded by*

$$\sum_{t=1}^{T} u_L(x_t, y_t) \geq \max_{\pi:[n]\to[n]} \sum_{t=1}^{T} u_L(\pi(x_t), y_t) - r(T).$$

where $\pi(x_t)$ refers to the linear extension of π acting on the support of x_t.

An algorithm is a ***no-swap regret algorithm*** *if it is an $r(T)$-swap regret algorithm with $r(T) = o(T)$.*

Here the maximum is over all swap functions $\pi : [n] \to [n]$ (extended linearly to act on elements y_t of Δ_n). It is a fundamental result in the theory of online learning that both no-swap-regret algorithms and no-regret algorithms exist (see [9]).

Some no-regret algorithms have the property that at each round, they approximately best-respond to the historical sequence of losses interpreted as a mixed strategy. Following [7] and [13], we call such algorithms *mean-based algorithms*. Formally, we define mean-based algorithms as follows.

▶ **Definition 22** (Mean-Based Algorithm). *An algorithm $\mathcal{A}$ is* $\gamma(t)$-mean-based *if whenever $j, j' \in [m]$ satisfy*

$$\frac{1}{t}\sum_{s=1}^{t} u_L(j', y_s) - \frac{1}{t}\sum_{s=1}^{t} u_L(j, x_s) \geq \gamma(t),$$

then $x_{t,j} \leq \gamma(t)$ (i.e., if j is at least $\gamma(t)$ worse than some other action j' against the historical average action of the opponent, then the total probability weight on j must be at most $\gamma(t)$). A learning algorithm is mean-based *if it is $\gamma(t)$-mean-based for some $\gamma(t) = o(1)$.*

Many standard no-regret learning algorithms are mean-based, including Multiplicative Weights, Hedge, Online Gradient Descent, and others (see [7]).

Up to low order terms, best responses to no-swap regret algorithms in algorithm space are well understood: the optimizer can do no better than to play a static non-responsive strategy which at every round plays an identical action distribution that is very close (in any reasonable norm) to their optimal Stackelberg leader strategy in the stage game:

▶ **Lemma 23** ([13]). *If the learner plays a no-swap-regret algorithm, then there exists $\varepsilon = o_T(1)$ and a $o(T)$-additive best-response of the optimizer that involves playing a distribution D' in each round such that $||D - D'||_\infty \leq \varepsilon$ where D is the optimizer's leader Stackelberg strategy of the underlying stage game.*

4 Results in the Bertrand Model

4.1 Equilibria of the Stage Game

In this section, we focus on Nash and Stackleberg Equilibria of the stage game. We remind the reader that the Nash Equilibria of the stage game define which prices are competitive. In turn, this section identifies what the baseline competitive prices are in the Bertrand model (the equivalent result for the logit model can be found in the full version).

We begin by showing that in any Nash equilibrium of the Bertrand stage game, the resulting competitive prices must be at most $\frac{2}{k}$, implying in particular that competitive prices yield in total a vanishing fraction of the monopoly revenue. To prove this, we need to show that pricing high relative to the other seller's price distribution is a dominated strategy, in a robust sense. Intuitively, it is clear that the best-response to a deterministic price is to undercut your opponent. We show that the same idea works against distributions in the following Lemma that we use throughout this section. The proof is deferred to the full version.

▶ **Lemma 24.** *Consider one seller (say seller 2) in the Bertrand stage game playing a distribution d where the total weight placed on prices above x (where $x \geq \frac{3}{k}$) is equal to b, for $b \leq \frac{1}{24k}$. Then, for the other seller (seller 1), there is some price $x' < x$ such that $u_1(x', d) \geq u_1(x, d) + \frac{1}{24k^2}$.*

The above lemma has a simple corollary which will also be useful:

▶ **Corollary 25.** *In a Bertrand stage game in which one seller's distribution is d, pricing at* $p \geq max(supp(d))$ *is a strictly dominated strategy when* $\max(supp(d)) \geq \frac{3}{k}$.

We are now ready to show what constitutes competitive prices in the Bertrand game.

▶ **Lemma 26.** *All Nash Equilibria of the Bertrand stage game are supported on prices that are at most* $\frac{2}{k}$.

Proof. We will show this by induction:
Base Case: No Nash Equilibrium (NE) of this game have support on price 1. To see this, note that by Corollary 25, the price 1 is dominated for both sellers, and thus cannot be in the support of any NE.
Induction Hypothesis: If no NE has support on price $\frac{m}{k}$ or higher and m is an integer s.t. $m \geq 4$, then no NE can have support on price $\frac{m-1}{k}$. To see this, consider any NE, (d_1, d_2) such that neither distribution is supported on prices $\frac{m}{k}$ or higher. Since $max(supp(d_1)) \leq \frac{m-1}{k}$, where $\frac{m-1}{k} \geq \frac{3}{k}$, the best-response to it cannot be $\frac{m-1}{k}$ (the proof of this is deferred to the full version). Therefore d_2 cannot have support on price $\frac{m-1}{k}$. A similar argument holds for d_1, extending the induction hypothesis.
Therefore, no Nash Equilibria of the game have support on any prices above $\frac{2}{k}$. ◀

We observe a gap between the equilibrium prices of the stage game between the Nash and the Stackelberg equilibrium. The following lemma relies upon results we will show in Section 4.2.1, but we state it here for ease of organization.

▶ **Lemma 27.** *The Stackelberg Equilibria of the stage game leads to supra-competitive prices*

Proof. By Lemma 30, the optimizer can achieve utility $\Omega_k(T)$ against a no-swap regret learner (by playing a static strategy). [13] establish that the maximum possible average payoff achievable against a no-swap regret learner is upper bounded by the stage game Stackelberg leader value. Therefore, the stage game Stackelberg leader value is $\geq \Omega_k(1)$, and thus the resulting price is $\geq \Omega_k(1)$ (see Observation 19 which lower bounds the average price by the average payoff of either seller). ◀

4.2 Results for the Repeated Bertrand Game

We begin by considering the Stackelberg equilibria of the repeated Bertrand game. As we observed earlier, allowing unrestricted commitment to any possible algorithm opens the door for explicit usage of threats to maintain collusive prices. The following lemma shows that supra-competitive prices emerge out of such commitments, and is unsurprising.

▶ **Lemma 28.** *The Stackelberg Equilibrium of the repeated Bertrand game induces supra-competitive prices.*

The proof, which is based on the algorithm of [11] to find optimal algorithmic Stackelberg strategies in repeated games, is deferred to the full version. It is known that the resulting leader algorithm is "obviously" anti-competitive, in that it encodes an explicit threat for the follower. However, we show that supra-competitive prices arise from a vast array of algorithms which are not facially anti-competitive. In particular, we consider the class of mean-based no-regret algorithms and the class of no-swap regret algorithms, which are well-behaved in the following sense: for both classes of algorithms, when both sellers use any algorithm in the class, the prices converge to competitive prices.

4.2.1 Committing to No-Regret Algorithms Induces Supra-Competitive Prices

In this section, we will prove two key results pertaining to the repeated version of the game:

1. While playing against any no-regret learner, the optimizer can always guarantee themselves at least $\Omega_k(T)$ net payoff, even when restricted to static non-responsive strategies (recall: strategies which are oblivious to their opponent's behavior and simply play the same pricing distribution every day). In fact, they attain $\Omega_k(T)$ net payoff even by playing a uniform distribution over prices each day.
2. If an optimizer attains $\Omega_k(T)$ net payoff against a no-regret Learner, that no-regret Learner also obtains $\Omega_k(T)$ net payoff.

Taken together, these two results imply that, if the optimizer, optimizing in the environment defined by the learner's no-regret algorithm, does at least as they could do using a simple static and non-responsive strategy, the induced prices are supra-competitive. Furthermore, *both* the learner and the optimizer benefit from these high prices. Finally, we will show that, when the learner is playing a no-swap regret algorithm and the optimizer plays a static non-responsive best response, this is in fact an approximate *Nash* Equilibrium in algorithm space (where the approximation is a sublinear additive factor). We emphasize that despite the fact that the optimizer happens to be playing a static, non-responsive algorithm, this is a best response for them in all of (unrestricted) algorithm space. This result in fact holds true in all games not just pricing games – but in the context of pricing games, it gives an example of a Nash equilibrium with supra-competitive prices between two algorithms which do not explicitly encode threats.

To achieve these results, we must reason about an optimizer's payoff when playing against a no-regret learner. Consider a no-regret algorithm playing against a seller who plays the same distribution in each round. This fixes the expected cumulative payoff of each fixed action for the no-regret learner; and thus, by the no-regret guarantee, the learner must play a best-response for all but a diminishing fraction of rounds. This, in turn, has implications for the follower's aggregate payoff. We capture this relationship in the following lemma. Note that this lemma applies to all bimatrix repeated games, not just pricing games.

▶ **Lemma 29.** *Consider any T-round repeated game defined by a stage game G with non-negative payoffs. Against any learning algorithm with the no regret property, if the optimizer plays a static strategy s each round, the optimizer's expected payoff is at least $u_o(BR^*_\ell(s), s) \cdot T - o(T)$, where u_o is the optimizer's value in G and $BR^*_\ell(s)$ is the element of the learner's best response to s in G that minimizes $u_o(BR^*_\ell(s), s)$.*

Proof. Let the total regret of the learning algorithm be $r(T)$, and let $BR_\ell(s)$ be the set of all best responses to s from the learner. Furthermore, let $g = u_\ell(BR^*_{\ell(s)}, s) - \max_{y \in [m] \setminus BR_{\ell(s)}} u_\ell(y, s)$.

Let p^* be the average probability distribution played by the learning algorithm against s. Since there is no correlation between the two seller's randomness, the learner's average payoff is

$$u_\ell(p^*, s) \leq \mathbb{P}[p^* = x \in BR_\ell(s)] \cdot u_\ell(BR_\ell(s), s) + \mathbb{P}[p^* = x \notin BR_\ell(s)] \cdot (u_\ell(BR_\ell(s), s) - g)$$

As the learner's algorithm has average regret $\frac{r(T)}{T}$, we have that

$$\mathbb{P}[p^* = x \in BR_l(s)] \cdot u_\ell(BR_\ell(s), s) + \mathbb{P}[p^* = x \notin BR_\ell(s)] \cdot [u_\ell(BR_\ell(s), s) - g] \geq u_\ell(BR_\ell(s), s) - \frac{r(T)}{T}.$$

Therefore, $\mathbb{P}[p^* = x \notin BR_\ell(s)] \cdot g \leq \frac{r(T)}{T}$, which implies $\mathbb{P}[p^* = x \notin BR_\ell(s)] \leq \frac{r(T)}{Tg}$, and finally $\mathbb{P}[p^* = x \in BR_\ell(s)] \geq 1 - \frac{r(T)}{Tg}$.

This allows us to lower bound the payoff of the optimizer (where the first inequality implicitly uses the fact that the both sellers always get non-negative payoff) –

$$\begin{aligned}
u_o(p^*, s) &\geq \mathbb{P}[p^* = x \in BR_\ell(s)] \cdot u_o(BR_\ell^*(s), s) \\
&\geq \left(1 - \frac{r(T)}{Tc}\right) u_o(BR_\ell^*(s), s) \\
&= u_o(BR_\ell^*(s), s) - \frac{r(T)}{T} \cdot \frac{u_o(BR_\ell^*(s), s)}{g} \\
&= u_o(BR_\ell^*(s), s) - \frac{o(T)}{T} \cdot \frac{u_o(BR_\ell^*(s), s)}{g} \\
&= u_o(BR_\ell^*(s), s) - \frac{o(T)}{T}.
\end{aligned}$$

◀

We are now ready to show that if an optimizer's only goal is maximizing their own payoff, and they do even a passable job, they will get high payoff against a no-regret learner. In fact, they can do this even by performing the extremely simple static strategy of pricing uniformly randomly every day. Recall that by Observation 19, the average payoff of the optimizer lower bounds the average price. Therefore, as long as the optimizer is optimizing in the environment defined by the learner's no regret algorithm, there is a wide range of optimizer behaviors which induce supra-competitive prices. This includes all optimizers that perform better than static random pricing, ranging from approximate optimization over static responses to exact optimal dynamic responses.

▶ **Lemma 30.** *The optimizer can get payoff at least $\Omega_k(T)$ against any no-regret learner in the Bertrand model by playing a static strategy of uniformly random pricing. Formally, for any no-regret algorithm $\mathcal{A}^{noreg}$ and the static uniformly random algorithm $\mathcal{A}^r$, $U_o(\mathcal{A}_l^{noreg}, \mathcal{A}_o^r) \geq \Omega_k(T)$.*

Proof. Let r be the uniformly random distribution over prices. By Lemma 29, the optimizer's expected payoff is at least $u_2(BR_\ell^*(r), r) \cdot T - o(T)$.

First, we will prove that any best-response of the learner $BR_\ell(r) \geq \frac{1}{5}$. To see this, note that by pricing at $\frac{1}{2}$ against r, the learner's payoff would be at least

$$\begin{aligned}
\mathbb{P}\left[r > \frac{1}{2}\right] \cdot \frac{1}{2} = \mathbb{P}\left[r \geq \lfloor \frac{1}{2} + \frac{1}{k} \rfloor\right] \cdot \frac{1}{2} \\
&= \frac{1 - \lfloor \frac{1}{2} + \frac{1}{k} \rfloor}{1 - \frac{1}{k}} \cdot \frac{1}{2} \\
&\geq \left(1 - (\frac{1}{2} + \frac{1}{k})\right) \cdot \frac{1}{2} \\
&\geq \frac{\frac{1}{2} - \frac{1}{k}}{2} \\
&\geq \frac{1}{4} - \frac{1}{2k} \\
&\geq \frac{1}{4} - \frac{1}{20} = \frac{1}{5}
\end{aligned}$$

As the learner can get payoff at least $\frac{1}{5}$, their best response price will always be at least $\frac{1}{5}$, as otherwise they would always get payoff strictly less than $\frac{1}{5}$.

We now invoke Lemma 29 to lower bound the payoff of the optimizer. The worst learner best-response to the uniform distribution, from the perspective of the optimizer, must be bounded below by $\frac{1}{5}$ i.e. the optimizer must select a price that is at least $p^* := \frac{\lfloor \frac{k}{5} \rfloor}{k}$ [3]. Additionally, the optimizer's payoff is monotone increasing in the learner's price. Thus the optimizer gets total payoff at least $T \cdot u_o(r, p^*) - o(T)$. Analyzing this gives us the desired result –

$$\begin{aligned} u_o(r, p^*) \cdot T - o(T) &\geq \mathbb{E}[r | r < p^*] \cdot \mathbb{P}[r < p^*] \cdot T - o(T) \\ &\geq \mathbb{E}\left[r | r \leq p^* - \frac{1}{k}\right] \cdot \mathbb{P}\left[r \leq p^* - \frac{1}{k}\right] \cdot T - o(T) \end{aligned}$$

$\mathbb{E}[r | r <= p^* - \frac{1}{k}]$ is exactly $\frac{p^*}{2}$ (the average of $\frac{1}{k}$ and $p^* - \frac{1}{k}$). Substituting p^* results in $\frac{p^*}{2} = \frac{\lfloor \frac{k}{5} \rfloor}{10}$, this is lower bounded by $\frac{2}{25}$ (assuming $k > 20$). On the other hand, we can rewrite $\mathbb{P}\left[r \leq p^* - \frac{1}{k}\right]$ (by substituting p^*) as $\frac{\lfloor \frac{k}{5} \rfloor - 1}{k}$. This term is monotone increasing in k and lower bounded by $\frac{3}{25}$ (assuming $k > 20$). Thus, the optimizer payoff is lower bounded by $\frac{6T}{625} - o(T)$, completing the proof. ◀

Since by Observation 19 the average payoff of the optimizer lower bounds the average price, we have already shown that any no-regret algorithm induces supra-competitive prices against any optimizer who performs better than random pricing. We will now go on to show that the learner deploying the no-regret algorithm benefits from these supra-competitive prices as well.

▶ **Theorem 31.** *For any no-regret learner algorithm $\mathcal{A}_l^{noreg}$, and for any optimizer algorithm $\mathcal{A}_o$, if*

$$U_o(\mathcal{A}_l^{noreg}, \mathcal{A}_o) \geq \Omega_k(T)$$

then

$$U_l(\mathcal{A}_l^{noreg}, \mathcal{A}_o) \geq \Omega_k(T)$$

Proof. Let $U_o(\mathcal{A}_l^{noreg}, \mathcal{A}_o) \geq c \cdot T$ for some constant c between 0 and 1, independent of k.

Note that, in order to achieve c payoff on average, the optimizer must have priced at or above $\frac{c}{2}$ with average probability (over the rounds) at least $\frac{c}{2}$. Assume for contradiction that the optimizer does not do this. Then, we can upper bound their utility by assuming they capture demand in all rounds:

$$\begin{aligned} U_o(\mathcal{A}_l^{noreg}, \mathcal{A}_o) &\leq \mathbb{P}\left[p_o \geq \frac{c}{2}\right] \cdot 1 + \left(1 - \mathbb{P}\left[p_o \geq \frac{c}{2}\right]\right) \cdot \left(\frac{c}{2}\right) \\ &< \frac{c}{2} + \left(1 - \frac{c}{2}\right) \cdot \frac{c}{2} \\ &= c - \frac{c^2}{4} \end{aligned}$$

This is strictly less than c, and thus by contradiction, the optimizer must have priced at or above $\frac{c}{2}$ with frequency at least $\frac{c}{2}$.

[3] By construction, p^* is in the set of prices available to the seller

Given this, one fixed action that the learner could have taken is to price at $\frac{c}{2}$. Then,

$$U_l(\mathcal{A}_l^{noreg}, \mathcal{A}_o) \geq \frac{1}{2}\mathbb{P}\left[p_o \geq \frac{c}{2}\right] \cdot \left(\frac{c}{2}\right)$$
$$\geq \frac{1}{2} \cdot \frac{c}{2} \cdot \left(\frac{c}{2}\right) = \frac{c^2}{8}$$

By the no-regret guarantee of the learner, therefore, we have that $U_l(\mathcal{A}_l^{noreg}, \mathcal{A}_o) \geq \frac{c^2 T}{8} - o(T) = \Omega_k(T)$. ◀

We can now combine these results, to show that, if the learner deploys any no-regret algorithm in the Bertrand model and the optimizer responds via any strategy which gets them payoff at least that of static random pricing, then prices are supra-competitive and both the learner and the optimizer get a constant fraction of the profits.

▶ **Theorem 32.** *In a Bertrand repeated game, for any no-regret learner algorithm $\mathcal{A}_l^{noreg}$ and any optimizer algorithm $\mathcal{A}_o$ such that*

$$U_o(\mathcal{A}_l^{noreg}, \mathcal{A}_o) \geq U_o(\mathcal{A}_l^{noreg}, \mathcal{A}_o^r), \tag{5}$$

where $\mathcal{A}_o^r$ is the static uniformly random algorithm, we have

$$U_o(\mathcal{A}_l^{noreg}, \mathcal{A}_o) = \Omega_k(T) \tag{6}$$

$$U_l(\mathcal{A}_l^{noreg}, \mathcal{A}_o) = \Omega_k(T) \tag{7}$$

Further, the average price is $\Omega_k(1)$.

Proof. By Lemma 30, the optimizer gets payoff at least $\Omega_k(1) \cdot T$ against any no-regret algorithm by playing static random pricing. Therefore, the optimizer's payoff utilizing any strategy which is better than static random pricing also gives them a payoff of at least $\Omega_k(1) \cdot T$. Furthermore, by Theorem 31, this implies that the learner also gets a payoff of at least $\Omega_k(1) \cdot T$. Finally, by Observation 19, this implies that the average price is $\Omega_k(1)$. ◀

We show an equivalent result for the more general Multinomial-logit model in the full version of the paper. Note that despite our treatment of the problem in a sequential play setting, neither sequential play nor commitment power is necessary for our results. In fact, in the game where leader and optimizer pick their algorithm simultaneously, if one player plays any no-swap regret algorithm and their opponent plays a static strategy corresponding to the Stackelberg leader distribution of the stage game, then this forms a Nash equilibrium in algorithm space:

▶ **Theorem 33.** *There exists an $o(T)$-approximate Nash Equilibrium in algorithm space in any repeated game consisting of a no-swap regret algorithm for the leader and a static, non-responsive algorithm for the follower.*

Proof. In particular, such an equilibrium exists between any algorithm with total swap regret $r(T) = o(T)$ and the static algorithm which plays (something very close to) the Stackelberg leader strategy of the stage game each round. To show that this is true, we will show that each is an $\frac{o(T)}{T}$-approximate best response to the other.

First, we will show that a particular distribution (which is ε-close to the Stackelberg leader strategy of the stage game) is near-optimal against any No-Swap Regret algorithm (NSR) when played on every round. By Lemma 23, there exists a distribution D' such that $||D - D'||_\infty \leq \varepsilon$, where D is the static Stackelberg leader strategy and D' is a $o(T)$-best response to any algorithm with sublinear swap regret $r(T)$.

It remains to show that NSR is optimal against this distribution. However, NSR is optimal against any static distribution played on every round. The best response against any algorithm playing a static distribution is to play (one of) the best-response action(s) each day. Therefore, the gap between the average payoff of the optimal response and the average payoff attained by the NSR algorithm is bounded by the average external regret of the NSR algorithm, which is at most $r(T) = o(T)$. ◀

Here, notably, both players are "succeeding in optimizing" since they are playing best responses in algorithm space against one another, neither is using an algorithm which encodes threats, and yet the outcome is near monopoly prices.

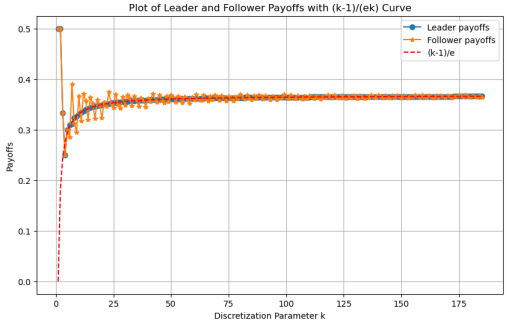
5 Numerical Investigation of Constants

We showed in Lemma 30 that the optimizer can get a payoff of $\Omega_k(T)$ against any no-regret learner, just by playing the uniform distribution over their prices. An implication is that they would do at least as well with their optimal static strategy, which was enough for our results which are stated in asymptotic notation. While we did not get attractive constants via this simple approach, we numerically verified that the optimal static optimizer distribution against a no-regret algorithm, which is the Stackelberg leader strategy for a number of values of k, results in average prices of more than $\frac{2}{3}$ (in fact approximately $\frac{2(k-1)}{ek}$). [12] give an efficient algorithm for computing the Stackelberg equilibrium of a two player game, which we use to numerically compute the Stackelberg leader strategy as well as the payoff of the Stackelberg leader, which will approximately (up to subconstant in T additive error) be the average payoff of the optimizer repeatedly playing the Stackelberg leader strategy (using Lemma 29). Our numerical results show that the Stackelberg value of the leader (as well as the follower) is approximately $\frac{k-1}{ek}$ for a range of values of k in Figure 1a. By Observation 19, if the follower optimizes over the space of static strategies, the average price for the buyer will be $\frac{2(k-1)}{ek} \geq \frac{2}{3}$. We note that while our experiments show these prices are achievable by non-responsive optimizers for all values of k from 1 to 200, it remains interesting future work to prove that this holds for all k. Finding a closed form analytical expression for the Stackelberg leader strategy of the stage game is an open problem and would likely help in proving the conjecture for all k.

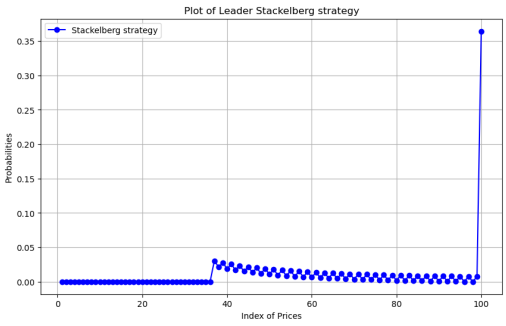
We also show the Stackelberg strategy for $k = 100$ in Figure 1b. Our numeric results also indicate that every price between 36/100 to 98/100 is (tied for being) a best response for the follower given the optimal leader commitment[4].

The code for these experiments can be found at Code Repository.

[4] In the Stackelberg Equilibrium problem, the follower typically tie-breaks in favor of the leader, in our application the optimizer adds some infinitesimal extra probability mass on price 99/100 to ensure that no-regret learner learns a unique best-response of 98/100 (which is the best outcome for the optimizer).



(a) Stackelberg Leader/Follower Payoffs for varying k.



(b) Stackelberg Leader Strategy for $k = 100$.

6 Discussion and Conclusion

Defining anti-competitive behavior is delicate. Monopoly like prices can arise in a number of scenarios: a *failure to optimize* (if e.g. players are simply not best responding to one another) as well as through *collusion* (which has been interpreted as requiring explicit *threats* when speaking of algorithmic collusion [8, 16]). Past work [17, 10] have proposed no-swap-regret algorithms as reasonable competitive algorithms in pricing scenarios, on the basis that they converge to competitive prices when both parties use them, and they seem both to successfully optimize and not to encode threats.

Our results complicate this picture. We show that if the first entrant into a market deploys a pricing algorithm with the no-swap-regret guarantee, then this very strongly incentivizes the next entrant to deploy an algorithm that will lead to supra-competitive prices. In fact, *anything the second entrant does* that obtains them profit at least that of a random pricing strategy will inevitably lead to supra-competitive prices. Moreover, this will not be at the expense of the first entrant – the no(-swap) regret learner will *also* enjoy supra-competitive revenue. And the algorithm of the second entrant can be entirely static and non-responsive, and therefore unable to encode threats. In fact, this phenomenon does not hinge on the sequential nature of play that we focus on and does not hinge on either player having commitment power. As we show, there is a Nash equilibrium of the game – in algorithm space – maintaining supra-competitive prices, which involves one player playing a no swap regret algorithm and the other playing a static pricing distribution. Both players are best responding to one another in the space of all pricing algorithms (without any restriction) and so neither player is failing to optimize – but neither are either of the players deploying threats. We suggest that this might require a reconsideration of what constitutes algorithmic collusion.

References

1 Ibrahim Abada and Xavier Lambin. Artificial intelligence: Can seemingly collusive outcomes be avoided? *Energy Engineering (Energy) eJournal*, 2020. URL: `https://api.semanticscholar.org/CorpusID:219341239`.

2 Ibrahim Abada, Xavier Lambin, and Nikolay Tchakarov. Collusion by mistake: Does algorithmic sophistication drive supra-competitive profits? *European Journal of Operational Research*, 318(3):927–953, 2024. `doi:10.1016/j.ejor.2024.06.006`.

3 Eshwar Ram Arunachaleswaran, Natalie Collina, Sampath Kannan, Aaron Roth, and Juba Ziani. Algorithmic collusion without threats, 2024. `doi:10.48550/arXiv.2409.03956`.

4 Stephanie Assad, Robert Clark, Daniel Ershov, and Lei Xu. Algorithmic pricing and competition: Empirical evidence from the german retail gasoline market. *Journal of Political Economy*, 132(3):723–771, 2024. doi:10.1086/726906.

5 Martino Banchio and Giacomo Mantegazza. Artificial intelligence and spontaneous collusion, 2023. arXiv:2202.05946.

6 Jean-Pierre Benoit, Vijay Krishna, et al. Finitely repeated games, 1984.

7 Mark Braverman, Jieming Mao, Jon Schneider, and Matt Weinberg. Selling to a no-regret buyer. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 523–538, 2018.

8 Emilio Calvano, Giacomo Calzolari, Vincenzo Denicolò, and Sergio Pastorello. Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, 110(10):3267–3297, 2020. doi:10.1257/aer.20190623.

9 Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

10 Sylvain Chassang and Juan Ortner. Regulating collusion. *Annual Review of Economics*, 15(Volume 15, 2023):177–204, 2023. doi:10.1146/annurev-economics-051520-021936.

11 Natalie Collina, Eshwar Ram Arunachaleswaran, and Michael Kearns. Efficient stackelberg strategies for finitely repeated games. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 643–651, 2023. doi:10.5555/3545946.3598695.

12 Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 82–90, 2006. doi:10.1145/1134707.1134717.

13 Yuan Deng, Jon Schneider, and Balasubramanian Sivan. Strategizing against no-regret learners. *Advances in neural information processing systems*, 32, 2019.

14 Sara Fish, Yannai A. Gonczarowski, and Ran I. Shorrer. Algorithmic collusion by large language models, 2024. doi:10.48550/arXiv.2404.00806.

15 Karsten T Hansen, Kanishka Misra, and Mallesh M Pai. Frontiers: Algorithmic collusion: Supra-competitive prices via independent algorithms. *Marketing Science*, 40(1):1–12, 2021. doi:10.1287/MKSC.2020.1276.

16 Joseph E Harrington. Developing competition law for collusion by autonomous artificial agents. *Journal of Competition Law & Economics*, 14(3):331–363, 2018.

17 Jason D. Hartline, Sheng Long, and Chenhao Zhang. Regulation of algorithmic collusion. In *Proceedings of the Symposium on Computer Science and Law*, CSLAW '24, pages 98–108, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3614407.3643706.

18 Timo Klein. Autonomous algorithmic collusion: Q-learning under sequential pricing. *The RAND Journal of Economics*, 52(3):538–558, 2021. doi:10.1111/1756-2171.12383.

19 Yoav Kolumbus and Noam Nisan. Auctions between regret-minimizing agents. In *Proceedings of the ACM Web Conference 2022*, WWW '22, pages 100–111. ACM, April 2022. doi:10.1145/3485447.3512055.

20 Rohit Lamba and Sergey Zhuk. Pricing with algorithms, 2022. arXiv:2205.04661.

21 Ryan Y. Lin, Siddhartha Ojha, Kevin Cai, and Maxwell F. Chen. Strategic collusion of llm agents: Market division in multi-commodity competitions, 2024. doi:10.48550/arXiv.2410.00031.

22 Michael L Littman and Peter Stone. A polynomial-time nash equilibrium algorithm for repeated games. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, pages 48–54, 2003. doi:10.1145/779928.779935.

23 Uri Nadav and Georgios Piliouras. No regret learning in oligopolies: Cournot vs. bertrand. In *International Symposium on Algorithmic Game Theory*, pages 300–311. Springer, 2010. doi:10.1007/978-3-642-16170-4_26.

24 Denis Nekipelov, Vasilis Syrgkanis, and Eva Tardos. Econometrics for learning agents. In *Proceedings of the sixteenth acm conference on economics and computation*, pages 1–18, 2015.

Rank Lower Bounds on Non-Local Quantum Computation

Vahid R. Asadi
University of Waterloo, Canada

Eric Culf
University of Waterloo, Canada

Alex May
Perimeter Institute for Theoretical Physics, Waterloo, Canada
University of Waterloo, Canada

Abstract

A non-local quantum computation (NLQC) replaces an interaction between two quantum systems with a single simultaneous round of communication and shared entanglement. We study two classes of NLQC, f-routing and f-BB84, which are of relevance to classical information theoretic cryptography and quantum position-verification. We give the first non-trivial lower bounds on entanglement in both settings, but are restricted to lower bounding protocols with perfect correctness. Within this setting, we give a lower bound on the Schmidt rank of any entangled state that completes these tasks for a given function $f(x, y)$ in terms of the rank of a matrix $g(x, y)$ whose entries are zero when $f(x, y) = 0$, and strictly positive otherwise. This also leads to a lower bound on the Schmidt rank in terms of the non-deterministic quantum communication complexity of $f(x, y)$. Because of a relationship between f-routing and the conditional disclosure of secrets (CDS) primitive studied in information theoretic cryptography, we obtain a new technique for lower bounding the randomness complexity of CDS.

2012 ACM Subject Classification Theory of computation → Quantum information theory

Keywords and phrases Non-local quantum computation, quantum position-verification, conditional disclosure of secrets

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.11

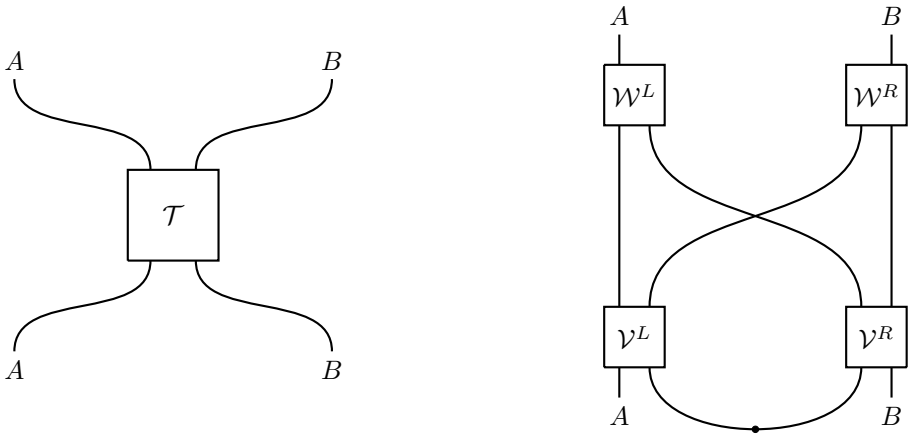
Funding *Alex May*: Supported by the Perimeter Institute for Theoretical Physics. Research at Perimeter Institute is supported in part by the Government of Canada through the Department of Innovation, Science and Economic Development Canada and by the Province of Ontario through the Ministry of Colleges and Universities.

Acknowledgements We thank Richard Cleve for helpful discussions during the development of this project. Harry Buhrman and François Le Gall suggested a connection between our bounds and non-deterministic rank, and consequently to non-deterministic quantum communication complexity.

1 Introduction

A non-local quantum computation (NLQC) replaces a local interaction with entanglement and a single simultaneous round of communication. See Figure 1. NLQC has applications in a number of areas, including quantum position-verification [19, 18, 20], Hamiltonian simulation [3], the AdS/CFT correspondence [21, 24, 22, 23], and information theoretic cryptography [2]. In all of these settings, we are interested in understanding the resources necessary to implement a given computation in the non-local form. Lower bounds on the entanglement necessary to complete a non-local computation are so far poorly understood but are central to these applications.


© Vahid R. Asadi, Eric Culf, and Alex May;
licensed under Creative Commons License CC-BY 4.0



(a) **(b)**

Figure 1 Local and non-local computations. a) A channel $\mathcal{T}_{AB\to AB}$ is implemented by directly interacting the input systems. b) A non-local quantum computation. The goal is for the action of this circuit on the AB systems to approximate the channel $\mathcal{T}_{AB\to AB}$.

Two important classes of non-local computations are f-routing and f-BB84, both initially studied in the context of quantum position-verification [18, 10]. These computations involve classical inputs of size n and $O(1)$ size quantum inputs. The local implementation of these computations involves computing a classical function f from the classical inputs, then doing $O(1)$ quantum operations conditioned on the outcome. A longstanding goal has been to prove that the non-local implementation of the same computation requires growing quantum resources, and in particular growing entanglement. Recently, a bound on the quantum system size was proven in [10], but this assumes a purified view that absorbs classical processing into the quantum part of the computation.[1] Another approach appears in [6], which bounds the number of quantum gates required in the non-local implementation. Here, we achieve the goal of lower bounding entanglement, at least as quantified by the Schmidt rank of the shared system, but at the cost of only considering protocols that complete f-routing or f-BB84 with perfect fidelity.[2]

It is a striking feature of non-local quantum computation that these mostly classical local computations require large quantum entanglement to implement non-locally. Beyond this, establishing this quantum-classical separation is relevant for quantum position-verification and for information theoretic cryptography. In a QPV scheme, we want to find operations that are as easy as possible for the honest player (who acts locally) to implement, but as hard as possible for the dishonest player (who implements an NLQC). Proving entanglement lower bounds for f-routing or f-BB84 gives a verification scheme where the honest player implements a nearly classical computation, while the dishonest player needs large quantum resources. This establishes the security of such QPV schemes under a bounded entanglement assumption. Our bound is only in the perfect setting and hence not immediately relevant to practical schemes, but represents a step towards proving entanglement lower bounds in the robust setting.[3]

[1] See [6] for more discussion of this point.

[2] Actually, we will see that in the case of f-routing we only need zero error on either $(x, y) \in f^{-1}(0)$ with bounded error for $(x, y) \in f^{-1}(1)$, or vice versa.

[3] Many entanglement lower bounds apply in the robust setting, see e.g. [25, 16, 23, 9], but none grow with the classical input size. We achieve growth with classical input size at the expense of robustness.

Also relevant to our work is the connection between NLQC and information theoretic cryptography. In information theoretic cryptography, we wish to understand the communication and randomness cost to establish information theoretic privacy in various cryptographic settings. A primitive known as conditional disclosure of secrets (CDS) has been identified as one of the simplest settings where one can hope to understand the cost of privacy. CDS was initially studied in the context of private information retrieval [17], has applications in secret sharing [4], and shares a number of connections to communication complexity and interactive proofs [5]. In [2], f-routing was shown to be closely related to CDS: lower bounds on entanglement cost in f-routing give lower bounds on randomness in CDS. This opens a new avenue for quantum information techniques to make contributions to classical information theoretic cryptography.

1.1 Summary of our results

An f-routing task is defined by a choice of Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. In the task, Alice, on the left, receives a quantum system Q of $O(1)$ size and a classical string $x \in \{0,1\}^n$. Bob, on the right, receives $y \in \{0,1\}^n$. Their goal is to bring Q to Alice if $f(x,y) = 0$, and to Bob if $f(x,y) = 1$. In the setting of non-local computation, Alice and Bob may only use shared entanglement and a single round of communication to accomplish this. A difficulty in accomplishing this task arises since quantum mechanics prevents the system in register Q from being copied, and hence apparently Alice is forced to decide alone whether to keep or send Q. Furthermore, she doesn't know where Q is needed until learning y, at which point the single round of communication has already passed.

Despite the difficulty, the f-routing task can be completed for arbitrary functions f, with the most efficient known protocols using $2^{\Theta(\sqrt{n \log n})}$ entanglement and communication for arbitrary functions [2], and efficient protocols known for some low complexity classes of functions [13, 12]. So far, only an $O(1)$ lower bound is known on entanglement in f-routing, as we review in Section A. Here, we prove a new lower bound on f-routing with one-sided perfection. We define a (ϵ_0, ϵ_1)-correct f-routing scheme as one which brings Q to the left with fidelity at least $1 - \epsilon_0$ when $(x,y) \in f^{-1}(0)$ and brings Q to the right with fidelity at least $1 - \epsilon_1$ when $(x,y) \in f^{-1}(1)$. Define $FR_0(f)$ as the log of the minimal Schmidt rank of any entangled state that suffices to complete f-routing with $\epsilon_0 = 0$, $\epsilon_1 > 0$ for the function f, and $FR_1(f)$ as the analogous quantity with $\epsilon_0 > 0, \epsilon_1 = 0$. We prove the bounds

$$\begin{aligned} \mathrm{FR}_0(f) &\geq \frac{1}{4} \log \operatorname{rank}(g|_f) \ , \\ \mathrm{FR}_1(f) &\geq \frac{1}{4} \log \operatorname{rank}(g|_{\neg f}) \ , \end{aligned} \tag{1}$$

where $g|_f(x,y)$ is a real-valued function that satisfies $f(x,y) = 0 \leftrightarrow g(x,y) = 0$. For some natural functions, this provides explicit strong lower bounds. For instance, taking $f = EQ$ to be the equality function, we are assured $g(x,y)$ is diagonal with non-zero entries on the diagonal and hence is of full rank, therefore $FR_0(EQ) \geq n/4$. Similarly, we obtain linear lower bounds on $FR_1(NEQ)$ where NEQ is the non-equality function.

As mentioned above, f-routing is related closely to the conditional disclosure of secrets (CDS) primitive. A CDS instance is defined by a choice of Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. In CDS, Alice and Bob receive inputs $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$ respectively, while Alice additionally holds a secret string $s \in \{0,1\}^k$. Alice and Bob share randomness, and send messages to a referee. Their goal is for the referee to be able to determine s if and only if $f(x,y) = 1$.

In [2], it was shown that the randomness cost of CDS for a function f upper bounds the entanglement cost of an f-routing scheme for the same function. Because of this, our lower bounds on entanglement in f-routing imply the same lower bounds on randomness in CDS. Because we only need one sided perfection for our bounds to apply, it is possible to bound CDS with imperfect correctness and perfect privacy (ppCDS), or perfect correctness but imperfect privacy (pcCDS)

$$\begin{aligned} \text{ppCDS}(f) &\geq \Omega(\log \operatorname{rank}(g|_f)) \ , \\ \text{pcCDS}(f) &\geq \Omega(\log \operatorname{rank}(g|_{\neg f})) \ . \end{aligned}$$

The expressions on the left hand side denote randomness costs. This provides a new technique for bounding randomness in CDS, which is proven using the quantum information perspective but can be phrased purely in terms of classical quantities.

We also prove lower bounds against the f-BB84 class of non-local quantum computations. An f-BB84 task is defined by a choice of Boolean function $f(x, y)$ with $x \in \{0,1\}^n$ given to Alice and $y \in \{0,1\}^n$ given to Bob. The system Q is in one of the BB84 states $\{H^{f(x,y)} |b\rangle\}$, with $b \in \{0,1\}$ and $f(x, y) \in \{0,1\}$. Alice and Bob should both output b.

The f-BB84 task can be completed for arbitrary functions f, with the most efficient known protocols using $2^{\Theta(n)}$ entanglement and communication for arbitrary functions [12].[4] We prove lower bounds on the log of the Schmidt rank of any resource state that suffices to complete f-BB84 with probability 1. In particular, the same bound as in the f-routing case applies,

$$\text{FBB84}(f) \geq \frac{1}{4} \log \operatorname{rank}(g|_f) \ ,$$

where again g is a function with zeros exactly at the zeros of f. In a practical setting, f-BB84 is of particular interest because it can be made loss-tolerant [1]. Unfortunately, our bound does not carry over to the noisy setting, so is not of immediate practical value. Nonetheless, our bound is the first non-trivial entanglement bound on f-BB84. For f-BB84, the bound above leads to linear lower bounds for the equality, non-equality, set-disjointness, and greater-than functions.

We can better understand our lower bounds by relating them to communication complexity. In a quantum communication complexity protocol, Alice holds $x \in \{0,1\}^n$, Bob holds $y \in \{0,1\}^n$, and Alice and Bob communicate qubits to determine $f(x, y)$. The number of qubits of communication needed for Alice and Bob to output 1 with non-zero probability when $f(x, y) = 1$, and never output 1 when $f(x, y) = 0$ is known as the quantum non-deterministic communication complexity, $QNP^{cc}(f)$. Using the characterization of QNP^{cc} in terms of the non-deterministic rank [14, 15], we obtain

$$\begin{aligned} \log \operatorname{rank} g|_f &\geq (\text{QNP}^{\text{cc}}(f) - 1) \ , \\ \log \operatorname{rank} g|_{\neg f} &\geq (\text{coQNP}^{\text{cc}}(f) - 1) \ . \end{aligned}$$

This relates our lower bounds on CDS, f-routing, and f-BB84 to communication complexity.

[4] Note that [12] deals with the f-routing task, but their protocol is easily adapted to f-BB84.

2 Preliminaries

We let Ψ^+_{AB} denote the maximally entangled state on AB, so that

$$\left|\Psi^+\right\rangle_{AB} = \frac{1}{\sqrt{d}} \sum_{i=1}^{d} |i\rangle_A |i\rangle_B \ ,$$

and $\Psi^+_{AB} = |\Psi^+\rangle\langle\Psi^+|_{AB}$.

Given two density matrices ρ, σ, we define the fidelity by

$$F(\rho,\sigma) = \operatorname{tr}\sqrt{\sqrt{\sigma}\rho\sqrt{\sigma}} \ .$$

The fidelity is related to the one-norm distance by the Fuchs–van de Graff inequalities,

$$1 - F(\rho,\sigma) \leq \frac{1}{2}||\rho - \sigma||_1 \leq \sqrt{1-(F(\rho,\sigma))^2} \ .$$

The von Neumann entropy is defined by

$$S(A) = -\operatorname{tr}(\rho \log \rho) \ .$$

The quantum mutual information is defined by

$$I(A:B)_\rho = S(A)_\rho + S(B)_\rho - S(AB)_\rho \ ,$$

and the conditional mutual information is defined by

$$S(A|B)_\rho = S(AB)_\rho - S(B)_\rho \ .$$

We will also make use of the following statement of continuity of the conditional entropy [26],

▶ **Theorem 1** ([26]). *Suppose that*

$$\frac{1}{2}||\rho_{AB} - \sigma_{AB}||_1 \leq \epsilon \ ,$$

and let $h(x) = -x\log x - (1-x)\log(1-x)$ *be the binary entropy function. Then*

$$|S(A|B)_\rho - S(A|B)_\sigma| \leq 2\epsilon \log d_A + (1+\epsilon) h\left(\frac{\epsilon}{1+\epsilon}\right) \ .$$

We will also make use of the following observation.

▶ **Observation 2.** *Let* ρ_{AB} *be a density matrix on the* AB *system,* d *be the dimension of subsystem* A*, and* ρ_B *be the marginal of* ρ_{AB} *on the subsystem* B*. Then we have*

$$\begin{aligned}
\operatorname{tr}\left(\rho_{AB} - \frac{\mathcal{I}_A}{d}\otimes\rho_B\right)^2 &= \operatorname{tr}\left(\rho_{AB}^2 + \frac{\mathcal{I}_A}{d^2}\otimes\rho_B^2 - 2\rho_{AB}(\frac{\mathcal{I}_A}{d}\otimes\rho_B)\right) \\
&= \operatorname{tr}\left(\rho_{AB}^2\right) + \frac{1}{d^2}\operatorname{tr}(\mathcal{I}_A)\operatorname{tr}\left(\rho_B^2\right) - \frac{2}{d}\operatorname{tr}_B(\rho_B^2) \\
&= \operatorname{tr}\left(\rho_{AB}^2\right) - \frac{1}{d}\operatorname{tr}\left(\rho_B^2\right) \\
&= \operatorname{tr}\left(\rho_{AB}^2\right) - \operatorname{tr}\left(\frac{\mathcal{I}_A}{d^2}\otimes\rho_B^2\right) \ .
\end{aligned}$$

2.1 Communication complexity

We need some definitions and results from communication complexity. Consider two parties, Alice and Bob, with Alice given input $x \in \{0,1\}^n$ and Bob given input $y \in \{0,1\}^n$. Alice and Bob exchange messages and Alice outputs a bit z. The *non-deterministic quantum communication complexity* $\mathrm{QNP}^{cc}(f)$, introduced in [14], is the minimal number of qubits Alice and Bob need to exchange to both output 1 with probability larger than zero when $f(x,y) = 1$ and both output 0 with probability 1 when $f(x,y) = 0$.

A matrix M is called a non-deterministic communication complexity matrix for f exactly when $M(x,y) \neq 0$ when $f(x,y) = 1$. The non-zero entries are allowed to be any complex numbers. The minimal rank of any non-deterministic communication complexity matrix for f is denoted by $nrank(f)$ and called the non-deterministic rank. We have the following result from [15].

▶ **Theorem 3** ([15]). *The non-deterministic rank and the non-deterministic quantum communication complexity are related by*

$$\mathrm{QNP}^{\mathrm{cc}}(f) = \lceil \log(\mathrm{nrank}(f)) \rceil + 1 \ .$$

3 Lower bounds from function rank

In this section, we first start by defining and analyzing the f-routing task, and then prove our result in this setting. Next, we will show similar lower bounds for f-BB84.

3.1 f-routing bounds from function rank

We begin with a definition of an f-routing task.

▶ **Definition 4.** *A **qubit f-routing** task is defined by a choice of Boolean function $f : \{0,1\}^{2n} \to \{0,1\}$, and a 2 dimensional Hilbert space $\mathcal{H}_Q$. Inputs $x \in \{0,1\}^n$ and system Q are given to Alice, and input $y \in \{0,1\}^n$ is given to Bob. Alice and Bob exchange one round of communication, with the combined systems received or kept by Bob labelled M' and the systems received or kept by Alice labelled M. Label the combined actions of Alice and Bob in the first round as $\mathcal{N}^{x,y}_{Q\to MM'}$. The qubit f-routing task is completed (ϵ_0, ϵ_1)-correctly on an input (x,y) if there exists a channel $\mathcal{D}^{x,y}_{M\to Q}$ such that,*

$$\text{when } f(x,y) = 0, \ \ F(\mathcal{D}^{x,y}_{M\to Q} \circ \mathrm{tr}_{M'} \circ \mathcal{N}^{x,y}_{Q\to MM'}(\Psi^+_{\bar{Q}Q}), \Psi^+_{\bar{Q}Q}) \geq 1 - \epsilon_0 \ ,$$

and there exists a channel $\mathcal{D}^{x,y}_{M'\to Q}$ such that

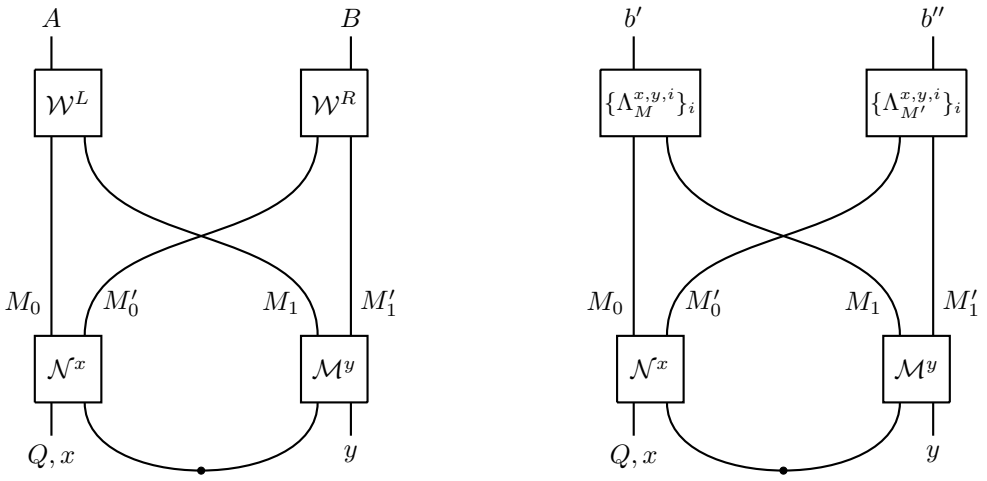
$$\text{when } f(x,y) = 1, \ \ F(\mathcal{D}^{x,y}_{M'\to Q} \circ \mathrm{tr}_{M} \circ \mathcal{N}^{x,y}_{Q\to MM'}(\Psi^+_{\bar{Q}Q}), \Psi^+_{\bar{Q}Q}) \geq 1 - \epsilon_1 \ .$$

In words, Alice can recover Q if $f(x,y) = 0$ and Bob can recover Q if $f(x,y) = 1$.

See the system labels shown in Figure 2a. Note that we label $M_0M_1 = M$, $M'_0M'_1 = M'$.

We will focus on f-routing with $\epsilon_0 = 0$, $\epsilon_1 \geq 0$ so that the protocol is perfectly correct on zero instances of $f(x,y)$, or with $\epsilon_0 > 0$ and $\epsilon_1 = 0$ so that the protocol is perfectly correct on 1 instances. We define the entanglement cost of an f-routing protocol to be the log of the minimal Schmidt rank of any resource system that can be used to perform the f-routing task. In notation, we define $\mathrm{FR}_0(f)$ to be entanglement cost for f-routing with $\epsilon_0 = 0$, $\epsilon_1 = 0.05$, $\mathrm{FR}_1(f)$ to be the entanglement cost when $\epsilon_0 = 0.05$, $\epsilon_1 = 0$.

Towards a lower bound in this setting, we first observe that $\rho_{\bar{Q}M'} = \rho_{\bar{Q}} \otimes \rho_{M'}$ iff $f(x,y) = 0$. We show this in the following lemma.



(a) **(b)**

Figure 2 a) General NLQC implementing a f-routing position-verification scheme. The first round operations are quantum channels that depend on the inputs $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$. In the second round, Alice and Bob apply channels mapping to qubit systems A, B. If $f(x,y) = 0$ A should be maximally entangled with the reference $\bar{Q}$. If $f(x,y) = 1$ then B should be maximally entangled with R. b) Non-local quantum computation implementing a f-BB84 position-verification scheme. The first round operations are quantum channels that depend on the inputs $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$. They communicate in one simultaneous exchange, and then apply measurements to their local systems. They succeed if $b = b' = b''$, with b determined by measuring a reference maximally entangled with Q.

▶ **Lemma 5.** *Suppose an f-routing protocol is perfectly correct on zero instances, and $\epsilon_1 < \epsilon_1^*$ correct on 1 instances, where ϵ_1^* depends on d_Q but is lower bounded by a constant. Then the mid-protocol state it produces, $\rho_{\bar{Q}M'}$, satisfies $\rho_{\bar{Q}M'} = \rho_{\bar{Q}} \otimes \rho_{M'}$ if and only if $f(x,y) = 0$.*

Proof. By correctness on 0 instances, we have that

$$\text{when } f(x,y) = 0, \quad F(\mathcal{D}^{x,y}_{M\to Q} \circ \mathrm{tr}_{M'} \circ \mathcal{N}^{x,y}_{Q\to MM'}(\Psi^+_{\bar{Q}Q}), \Psi^+_{\bar{Q}Q}) = 1 \ ,$$

hence $f(x,y) = 0$ implies Alice can produce a Bell state $|\Psi^+\rangle_{\bar{Q}Q}$. By data processing we have

$$2n = I(\bar{Q} : Q)_{\Psi^+_{\bar{Q}Q}} \leq I(\bar{Q} : M)_{\rho_{\bar{Q}M}} \ ,$$

but we also have the inequality

$$I(\bar{Q} : M)_\rho + I(\bar{Q} : M')_\rho \leq 2S(\bar{Q}) \ ,$$

so that $I(\bar{Q} : M') = 0$, which implies

$$\rho_{\bar{Q}M'} = \rho_{\bar{Q}} \otimes \rho_{M'} \ ,$$

as needed. Conversely, if $f(x,y) = 1$ we cannot have $\rho_{\bar{Q}M'} = \rho_{\bar{Q}} \otimes \rho_{M'}$. To see why, first recall that f-routing which is ϵ_1-correct on 1 instances has that there exists a family of channels $\{D^{x,y}_{M'\to Q}\}_{x,y}$ such that

$$\text{when } f(x,y) = 1, \quad F(\mathcal{D}^{x,y}_{M'\to Q} \circ \mathrm{tr}_{M} \circ \mathcal{N}^{x,y}_{Q\to MM'}(\Psi^+_{\bar{Q}Q}), \Psi^+_{\bar{Q}Q}) \geq 1 - \epsilon_1 \ .$$

Define

$$\sigma_{\bar{Q}Q} = \mathcal{D}^{x,y}_{M'\to Q}(\rho_{\bar{Q}M'}) \ ,$$

and use that by data processing,

$$I(\bar{Q}:M')_{\rho_{\bar{Q}M}} \geq I(\bar{Q}:Q)_{\sigma_{\bar{Q}Q}} \ .$$

But then by the Fuchs–van de Graff inequalities $\sigma_{\bar{Q}Q}$ is close in trace distance to $\Psi^+_{\bar{Q}Q}$,

$$\frac{1}{2}||\Psi^+_{\bar{Q}Q} - \sigma_{\bar{Q}Q}||_1 \leq \sqrt{2\epsilon_1 - \epsilon_1^2} \equiv \mu \ ,$$

and by continuity of the conditional entropy,

$$\begin{aligned} I(\bar{Q}:Q)_{\Psi^+} - I(\bar{Q}:Q)_{\sigma_{\bar{Q}Q}} &= S(\bar{Q}|Q)_{\Psi^+} - S(\bar{Q}|Q)_\sigma \\ &\leq 2\mu \log d_{\bar{Q}} + (1+\mu)h\left(\frac{\mu}{1+\mu}\right) \ , \end{aligned}$$

so that

$$2\log d_{\bar{Q}} - 2\mu \log d_{\bar{Q}} + (1+\mu)h\left(\frac{\mu}{1+\mu}\right) \leq I(\bar{Q}:Q)_\sigma \ .$$

We have that σ is not a tensor product whenever the left hand side of the equation above is strictly positive. For $d_Q = 2$, this occurs for $\epsilon_1 > \epsilon_1^* \approx 0.08$, and ϵ_1^* approaches 1 as $d_Q \to \infty$. ◀

Next, we define what we call a *structure function* for a protocol. The structure function is computed from the mid-protocol density matrix $\rho_{\bar{Q}MM'}$, and (as we will see) captures the zeros in $f(x,y)$.

▶ **Definition 6.** *Given an f-routing protocol with mid-protocol density matrix $\rho_{\bar{Q}MM'}$, define the **structure function** $g(x,y)$ according to*

$$g(x,y) = \mathrm{tr}\left(\rho_{\bar{Q}M'} - \frac{\mathcal{I}}{d_{\bar{Q}}} \otimes \rho_{M'}\right)^2 \ .$$

We claim that $g(x,y)$ captures some aspect of the structure in the function f which must be present in a correct f-routing protocol. More concretely we have the following.

▶ **Lemma 7.** *In a perfectly correct f-routing protocol, the structure function $g(x,y)$ is zero if and only if $f(x,y) = 0$.*

Proof. This follows because $g(x,y) = 0$ is zero if and only if $\rho_{\bar{Q}M'} = \frac{\mathcal{I}}{d_{\bar{Q}}} \otimes \rho_{M'}$, and Lemma 5 shows we have this tensor product form if and only if $f(x,y) = 0$. ◀

Our next job is to relate the function $g(x,y)$ to the entanglement available to Alice and Bob. We prove the following lemma.

▶ **Lemma 8.** *An f-routing protocol that uses a resource system with Schmidt rank d_E has a structure function of the form*

$$g(x,y) = \sum_{I=1}^{d_E^4} f_I(x) f'_I(y) \ .$$

Proof. From the general form of a non-local quantum computation protocol, the density matrix $\rho_{\bar{Q}M_0'M_1'}(x,y)$ can be expressed as

$$\rho_{\bar{Q}M_0'M_1'} = \mathcal{N}^x_{QL\to M_0'} \otimes \mathcal{M}^y_{\bar{Q}\to M_1'}(\Psi^+_{\bar{Q}Q} \otimes |\Psi\rangle\langle\Psi|_{LR}) \ .$$

We will write $|\Psi\rangle_{LR}$ in the Schmidt basis,

$$|\Psi\rangle_{LR} = \sum_{i=1}^{d_E} |i\rangle_L |i\rangle_R \ ,$$

with un-normalized vectors $|i\rangle_L, |i\rangle_R$. Then we get

$$\begin{aligned}\rho_{\bar{Q}M_0'M_1'} &= \sum_{i,j=1}^{d_E} \mathcal{N}^x_{QL\to M_0'}(\Psi^+_{Q\bar{Q}} \otimes |i\rangle\langle j|_L) \otimes \mathcal{M}^y_{R\to M_1'}(|i\rangle\langle j|_R) \\ &= \sum_{i,j=1}^{d_E} A^{x,i,j}_{\bar{Q}M_0'} \otimes B^{y,i,j}_{M_1'} \ .\end{aligned}$$

We can also compute the trace over $\bar{Q}$ where we define A and B in the second line of the previous equation and get

$$\rho_{M_0'M_1'} = \sum_{i,j=1}^{d_E} A^{x,i,j}_{M_0'} \otimes B^{y,i,j}_{M_1'} \ .$$

Next, we compute $g(x,y)$. It is convenient to make use of Observation 2 to write

$$g(x,y) = \mathrm{tr}\left(\rho^2_{\bar{Q}M'} - \frac{\mathcal{I}}{d^2_{\bar{Q}}} \otimes \rho^2_{M'}\right) \ .$$

Inserting the forms of $\rho_{\bar{Q}M'}$ and $\rho_{M'}$ into this, we obtain

$$\begin{aligned}g(x,y) &= \sum_{i,j,i',j'=1}^{d_E} \mathrm{tr}\left(\left(A^{x,i,j}_{QM_0'}A^{x,i',j'}_{QM_0'} - \frac{\mathcal{I}}{d^2_Q} \otimes A^{x,i,j}_{M_0'}A^{x,i',j'}_{M_0'}\right) \otimes B^{y,i,j}_{M_1'}B^{y,i',j'}_{M_1'}\right) \\ &= \sum_{I=1}^{d_E^4} f_I(x) f_I'(y) \ ,\end{aligned}$$

as needed. ◀

We can view $g(x,y)$ as a matrix, and $f_I(x)$, $f_I'(y)$ as vectors so that the minimal number of terms appearing in this sum is the rank of the matrix $g(x,y)$. Thus we obtain the lower bound

$$\boxed{\mathrm{FR}_0(f) \geq \frac{1}{4}\log \mathrm{rank}\, g|_f \ .} \tag{2}$$

We know that g is zero and non-zero for the same inputs as f, for which we have given the $g|_f$ notation as a reminder. In some cases, this is enough to let us lower bound the rank of g, as we discuss below.

We can also notice that we can reverse the role of M and M', and assume perfect correctness on 1 instances, leading to a similar bound,

$$\boxed{\mathrm{FR}_1(f) \geq \frac{1}{4}\log \mathrm{rank}\, g|_{\neg f} \ ,} \tag{3}$$

where $\neg f$ is the negation of f.

We can also rephrase the above bounds in terms of the non-deterministic quantum communication complexity. Observe that

$$\operatorname{rank} g|_f \geq \operatorname{nrank}(f) \ . \tag{4}$$

Because the non-deterministic rank considers matrices with arbitrary complex values where $f(x,y) = 1$, while $g|_f$ has the further constraint that it has positive real values when $f(x,y) = 1$, it is possible there are functions for which $\operatorname{rank} g|_f > \operatorname{nrank}(f)$. We have not looked for such examples.

Using Equation (4) and Theorem 3, we obtain

$$\mathrm{FR}_0(f) \geq \frac{1}{4}(\mathrm{QNP}^{\mathrm{cc}}(f) - 1) \ ,$$
$$\mathrm{FR}_1(f) \geq \frac{1}{4}(\mathrm{coQNP}^{\mathrm{cc}}(f) - 1) \ .$$

Further, for f-routing which has two sided perfection, we have

$$\mathrm{pFR}(f) \geq \frac{1}{4}\max\{\mathrm{QNP}^{\mathrm{cc}}(f), \mathrm{coQNP}^{\mathrm{cc}}(f)\} - 1 \ .$$

Or, labelling the set of function families which can be f-routed on with polylogarithmic entanglement by pFR, we have

$$\mathrm{pFR} \subseteq \mathrm{QNP}^{\mathrm{cc}} \cap \mathrm{coQNP}^{\mathrm{cc}} \ .$$

In the classical setting, $\mathrm{NP}^{\mathrm{cc}} \cap \mathrm{coNP}^{\mathrm{cc}} = \mathrm{P}^{\mathrm{cc}}$ [8], but (to our knowledge) it is not known if the quantum analogue holds.

3.2 f-BB84 bounds

▶ **Definition 9.** *A **qubit f-BB84** task is defined by a choice of Boolean function $f : \{0,1\}^{2n} \to \{0,1\}$, and a 2 dimensional Hilbert space $\mathcal{H}_Q$. Inputs $x \in \{0,1\}^n$ and system Q are given to Alice, and input $y \in \{0,1\}^n$ is given to Bob. The Q system is in the maximally entangled state with a reference $\bar{Q}$. Alice and Bob exchange one round of communication, with the combined systems received or kept by Bob labelled M and the systems received or kept by Alice labelled M'. Define projectors*

$$\Pi^{q,b} = H^q \left|b\right\rangle\!\left\langle b\right| H^q \ .$$

The qubit f-BB84 task is completed ϵ-correctly on input (x,y) if there exist POVM's $\{\Lambda_M^{x,y,0}, \Lambda_M^{x,y,1}\}$, $\{\Lambda_{M'}^{x,y,0}, \Lambda_{M'}^{x,y,1}\}$ such that,

$$\operatorname{tr}\left(\Pi_{\bar{Q}}^{f(x,y),b} \otimes \Lambda_M^{x,y,b} \otimes \Lambda_M^{x,y,b} \rho_{\bar{Q}MM'}\right) \geq 1 - \epsilon \ .$$

That is, Alice and Bob succeed when they both obtain the same outcome as the referee.

We will focus here on perfectly correct f-BB84, where $\epsilon = 0$.

As in the analysis of f-routing, we look for a function of the density matrix prepared by the protocol which captures the structure of the function $f(x,y)$. We define this next.

▶ **Definition 10.** *Given a f-BB84 protocol with mid-protocol density matrix $\rho_{\bar{Q}MM'}$, define the structure function by first defining the density matrices*

$$\rho_M^0 = \left\langle 0\right|_{\bar{Q}} \rho_{\bar{Q}M} \left|0\right\rangle_{\bar{Q}} \ ,$$
$$\rho_M^1 = \left\langle 1\right|_{\bar{Q}} \rho_{\bar{Q}M} \left|1\right\rangle_{\bar{Q}} \ .$$

These are the post measurement states on M when the referee measures R in the computational basis, and finds outcome $b = 0, 1$. We similarly define $\rho^0_{M'}$ and $\rho^1_{M'}$. Then define

$$g(x,y) = \mathrm{tr}\left(\rho^0_M \rho^1_M\right) + \mathrm{tr}\left(\rho^0_{M'} \rho^1_{M'}\right) \ .$$

We show this has the same zeros as $f(x,y)$ next.

▶ **Lemma 11.** *In a perfectly correct f-BB84 protocol, the structure function $g(x,y)$ is zero if and only if $f(x,y)$ is zero.*

Proof. We first establish the following fact: $g(x,y) = 0$ if and only if both ρ^0_M and ρ^1_M have non-overlapping support and $\rho^0_{M'}$ and $\rho^1_{M'}$ have non-overlapping support. To see this, first assume $g(x,y) = 0$. Then diagonalize ρ^0_M and ρ^1_M so that

$$\rho^0_M = \sum_i \lambda_i \left|\psi_i\right\rangle\!\left\langle\psi_i\right| \ ,$$
$$\rho^1_M = \sum_j \mu_j \left|\phi_j\right\rangle\!\left\langle\phi_j\right| \ ,$$

with $\lambda_i, \mu_j > 0$ and we have from $g(x,y) = 0$ that $\mathrm{tr}\left(\rho^0_M \rho^1_M\right) = 0$, so that

$$0 = \mathrm{tr}\left(\rho^0_M \rho^1_M\right) = \sum_{i,j} \lambda_i \mu_j \left|\left\langle\phi_j|\psi_i\right\rangle\right|^2 \ .$$

But this occurs if and only if

$$\forall i, j, \quad \left\langle\phi_j|\psi_i\right\rangle = 0 \ ,$$

so that ρ^0_M and ρ^1_M have non-overlapping support, as needed. We can similarly conclude $\rho^0_{M'}$ and $\rho^1_{M'}$ have non-overlapping support. For the converse, a direct calculation shows both $\mathrm{tr}\left(\rho^0_M \rho^1_M\right)$ and $\mathrm{tr}\left(\rho^0_{M'} \rho^1_{M'}\right)$ are 0.

Now we use this fact to relate $g(x,y) = 0$ and $f(x,y) = 0$. First, suppose that $f(x,y) = 0$. Then, the referee measures in the computational basis and so after the referee's measurement, Alice holds either ρ^0_M or ρ^1_M depending on the referee's measurement outcome. By assumption she can determine the measurement outcome perfectly, so she must be able to distinguish ρ^0_M and ρ^1_M perfectly. Thus ρ^0_M and ρ^1_M have orthogonal support. Considering the same line of reasoning but now considering Bob, whose post-measurement state is $\rho^0_{M'}$ or $\rho^1_{M'}$, these density matrices too must have orthogonal support. But then from the fact above we have $g(x,y) = 0$.

Next suppose that $f(x,y) = 1$. Then the referee measures in the Hadamard basis, so Alice's post-measurement state is one of

$$\rho^{\pm}_M = \left\langle\pm\right|_{\bar{Q}} \rho_{\bar{Q}M} \left|\pm\right\rangle_{\bar{Q}} \ .$$

For Alice to be able to determine the referee's measurement outcome, ρ^+_M and ρ^-_M must have orthogonal support, so that

$$0 = \mathrm{tr}\left(\rho^+_M \rho^-_M\right) \ . \tag{5}$$

We claim that this implies $\mathrm{tr}\left(\rho^0_M \rho^1_M\right) > 0$. To see why, note

$$\rho^{\pm}_M = \frac{1}{2}\left(\rho^0_M + \rho^1_M \pm \left\langle 0\right|_{\bar{Q}} \rho_{\bar{Q}M} \left|1\right\rangle_{\bar{Q}} \pm \left\langle 1\right|_{\bar{Q}} \rho_{\bar{Q}M} \left|0\right\rangle_{\bar{Q}}\right) \ .$$

Inserting this into Equation 5, we find that

$$\mathrm{tr}\big(\rho^0_M \rho^1_M\big) = \frac{1}{2}\left(\mathrm{tr}\big((\rho^0_M)^2\big) + \mathrm{tr}\big((\rho^1_M)^2\big)\right) > 0 \ .$$

Again, repeating the argument considering Bob's post-measurement state and his ability to determine the referee's measurement outcome, we also find that

$$\mathrm{tr}\big(\rho^0_{M'} \rho^1_{M'}\big) > 0 \ .$$

Thus, $g(x,y) > 0$ as needed. ◀

▶ **Lemma 12.** *An f-BB84 protocol that uses a resource system with Schmidt rank d_E has a structure function of the form*

$$g(x,y) = \sum_{I=1}^{2d_E^4} f_I(x) f'_I(y) \ .$$

Proof. From the general form of a non-local quantum computation protocol, the density matrix $\rho_{\bar{Q}M'_0M'_1}(x,y)$ can be expressed as

$$\rho_{\bar{Q}M'_0M'_1} = \mathcal{N}^x_{QL\to M'_0} \otimes \mathcal{M}^y_{R\to M'_1}(\Psi_{\bar{Q}Q} \otimes |\Psi\rangle\langle\Psi|_{LR}) \ .$$

We will write $|\Psi\rangle_{LR}$ in the Schmidt basis,

$$|\Psi\rangle_{LR} = \sum_{i=1}^{d_E} |i\rangle_L |i\rangle_R \ ,$$

with un-normalized vectors $|i\rangle_L, |i\rangle_R$. Then we get

$$\begin{aligned}\rho_{\bar{Q}M'_0M'_1} &= \sum_{i,j=1}^{d_E} \mathcal{N}^x_{QL\to M'_0}(\Psi^+_{\bar{Q}Q} \otimes |i\rangle\langle j|_L) \otimes \mathcal{M}^y_{R\to M'_1}(|i\rangle\langle j|_R) \\ &= \sum_{i,j=1}^{d_E} A^{x,i,j}_{\bar{Q}M'_0} \otimes B^{y,i,j}_{M'_1} \ .\end{aligned}$$

We can also project $\bar{Q}$ into $|b\rangle_{\bar{Q}}$, $b \in \{0,1\}$ to get $\rho^i_{M'}$,

$$\rho^b_{M'_0M'_1} = \sum_{i,j=1}^{d_E} A^{b,x,i,j}_{M'_0} \otimes B^{y,i,j}_{M'_1} \ .$$

A similar calculation gives ρ^b_M. Inserting these expressions into $g(x,y)$, we obtain an expression of the form

$$g(x,y) = \sum_{I=1}^{2d_E^4} f_I(x) f'_I(y) \ .$$

where the d_E^4 appears because each of the two trace terms in $g(x,y)$ comes with a product of two density matrices, each of which involves two sums running from 1 to d_E. We collect all of these sums into the index i appearing here, which now runs 1 to $2d_E^4$. ◀

Similarly to the case of f-routing, the above lemma implies that

$$\boxed{\mathrm{FBB84}(f) \geq \frac{1}{4}(\log \operatorname{rank} g|_f - 1)\ .} \tag{6}$$

where the notation $g|_f$ indicates as before that g is zero and non-zero on the same inputs as $f(x, y)$.

Also similarly to the case of f-routing, we can use Equation (4) and Theorem 3, we obtain

$$\mathrm{FBB84}(f) \geq \frac{1}{4}(\mathrm{QNP}^{\mathrm{cc}}(f) - 2)\ .$$

4 A new lower bound technique for CDS

It was recently understood [2] that f-routing is closely related to a classical information theoretic primitive known as conditional disclosure of secrets. As a consequence, our new lower bounds on f-routing imply lower bounds on CDS. In this work, we will informally describe these lower bounds, but see [7] for a more complete treatment.

The *conditional disclosure of secrets* primitive involves three parties, Alice, Bob, and the referee. Alice receives input $x \in \{0,1\}^n$, and Bob receives $y \in \{0,1\}^n$. Alice additionally holds a secret string z, and Alice and Bob share a random string r. Alice and Bob do not communicate, but each sends a message to the referee. The goal is for the referee to be able to learn z if and only if a function $f(x, y) = 1$, where the choice of the function f is known to all parties, and the referee knows both inputs x and y.

The quantum generalization of CDS, *conditional disclosure of quantum secrets* or CDQS, was recently defined [2]. In the quantum setting, Alice and Bob share a quantum state (rather than a random string) and can send quantum messages to the referee. The secret can now also be taken to be a quantum system Q rather than a classical string. The security and privacy conditions are similar to the classical case: correctness means that the referee can invert the map on Q applied by Alice and Bob when $f(x, y) = 1$, and security means the map applied by Alice and Bob is close to completely depolarizing when $f(x, y) = 0$. In general, we allow small correctness and privacy errors. See [2] for a formal definition.

We define the minimal number of qubits in Alice and Bob's shared resource system in a CDQS scheme for function f by $\mathrm{CDQS}(f)$. If we require the protocol to be perfectly correct we write $\mathrm{pcCDQS}(f)$, and when perfectly private we write $\mathrm{ppCDQS}(f)$. Similarly, for classical CDS, we use $\mathrm{CDS}(f)$ to denote the randomness cost of classical CDS. Again we modify this to $\mathrm{pcCDS}(f)$, $\mathrm{ppCDS}(f)$ for perfectly correct and perfectly private CDS randomness cost.[5]

In [2], it was shown that a good classical CDS protocol for a function f implies a good quantum CDS protocol for the same function. This leads to[6]

$$\mathrm{CDS}(f) = \Omega(\mathrm{CDQS}(f))\ ,$$

and the same relationships hold for the perfectly correct and perfectly private variants. Also in [2], it was shown that a good CDQS leads to a good f-routing protocol, and vice versa.

[5] The reader should be warned that the same notation is used to denote the *communication* cost in other papers on CDS.

[6] In fact, to obtain this we also need an amplification result which appears in [7]. These bounds are proved formally there.

This leads to

$$\mathrm{CDQS}(f) = \Theta(\mathrm{FR}(f)) \ ,$$

Also, the perfectly private and perfectly correct variants are related to f-routing with one-sided errors,

$$\mathrm{ppCDQS}(f) = \Theta(\mathrm{FR}_0(f)) \ ,$$
$$\mathrm{pcCDQS}(f) = \Theta(\mathrm{FR}_1(f)) \ .$$

This implies our lower bounds from Equation (2) and Equation (3) apply to CDS,

$$\mathrm{ppCDQS}(f) = \Omega(\log \operatorname{rank} g|_f) \ ,$$
$$\mathrm{pcCDQS}(f) = \Omega(\log \operatorname{rank} g|_{\neg f}) \ .$$

This also gives lower bounds in terms of $\mathrm{QNP}^{\mathrm{cc}}(f)$ and $\mathrm{coQNP}^{\mathrm{cc}}(f)$, and the upper bound can be written in terms of classical CDS. In this way, we can obtain the lower bound

$$\boxed{\mathrm{pcCDS}(f) \geq \Omega(\mathrm{coQNP}^{\mathrm{cc}}(f)) \ .}$$

This is similar to a lower bound obtained in [5],

$$\mathrm{pcCDS}(f) \geq \frac{1}{2}\mathrm{coNP}^{\mathrm{cc}}(f) \ .$$

The lower bound on perfectly private CDS randomness cost seems to be new,

$$\boxed{\mathrm{ppCDS}(f) \geq \Omega(\mathrm{QNP}^{\mathrm{cc}}(f)) \ .}$$

For perfect CDS, we can bound the randomness cost by the maximum of the $\mathrm{QNP}^{\mathrm{cc}}$ and $\mathrm{coNP}^{\mathrm{cc}}$ cost. If we let pCDS denote the set of families of functions for which CDS can be implemented with perfect correctness and perfect privacy using polylogarithmic randomness. Then we obtain

$$\mathrm{pCDS} \subseteq \mathrm{QNP}^{\mathrm{cc}} \cap \mathrm{coNP}^{\mathrm{cc}} \ .$$

5 Application to concrete functions

We detail lower bounds for equality, non-equality, greater than and less than, set-disjointness, and set-intersection below. We were not able to use our technique to lower bound inner product.

Our results are summarized succinctly in Figure 3.

Equality and non-equality

Choose $f(x, y)$ to be the equality function,

$$EQ(x, y) = \begin{cases} 0, & x \neq y \\ 1, & x = y \end{cases} \ .$$

Then $g(x, y)$ is zero except on the diagonal, which forces it to have full rank, so from Equation (2) and Equation (6)

$$\mathrm{FR}_0(EQ) \geq \frac{n}{4}, \qquad \mathrm{pFBB84}(EQ) \geq \frac{1}{4}(n-1) \ .$$

For the not-equals function, we can use bound from Equation (3) to obtain

$$\mathrm{FR}_1(NEQ) \geq \frac{n}{4} \ .$$

Function	pFR	FR_1	FR_0	FR
Equality	$\Theta(n)$	$\Theta(1)$	$\Theta(n)$	$\Theta(1)$
Non-Equality	$\Theta(n)$	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$
Inner-Product	$O(n)$, $\Omega(1)$	$O(n)$, $\Omega(1)$	$O(n)$, $\Omega(1)$	$O(n)$, $\Omega(1)$
Greater-Than	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$O(n)$, $\Omega(1)$
Set-Intersection	$\Theta(n)$	$\Theta(n)$	$O(n)$,$\Omega(1)$	$O(n)$, $\Omega(1)$
Set-Disjointness	$\Theta(n)$	$O(n)$,$\Omega(1)$	$\Theta(n)$	$O(n)$, $\Omega(1)$

Figure 3 Known upper and lower bounds on entanglement cost (quantified by the log Schmidt rank) in f-routing for natural functions. No non-trivial lower bounds on entanglement were known prior to this work. Linear upper bounds on all of these functions follow from [12, 13]. The $\Omega(1)$ lower bounds follow from Section A.

Greater and less than

Similarly, the "greater than" function,

$$GT(x,y) = \begin{cases} 0, & x < y \\ 1, & x \geq y \end{cases} .$$

is upper triangular with non-zero elements on the diagonal, so it also has full rank, and we obtain a linear lower bound.

$$\mathrm{FR}_0(GT) \geq \frac{n}{4}, \qquad \mathrm{pFBB84}(GT) \geq \frac{1}{4}(n-1) \ .$$

Further, because the negation of Greater-Than is also full rank, we can also bound fR_1,

$$\mathrm{FR}_1(GT) \geq \frac{n}{4} \ .$$

The same bounds hold for the "less than" function.

Set-Intersection and set-disjointness

Set disjointness is upper left triangular.[7] This implies it is full rank, therefore

$$\mathrm{FR}_0(DISJ) \geq \frac{n}{4}, \qquad \mathrm{pFBB84}(DISJ) \geq \frac{1}{4}(n-1) \ .$$

Since the negation of set intersection is set disjointness and hence of full rank, we also obtain

$$\mathrm{FR}_1(INT) \geq \frac{n}{4} \ .$$

6 Discussion

In this article, we have given the first non-trivial lower bounds on entanglement in the f-routing and f-BB84 tasks. For some natural functions, including equality, greater than, and set-disjointness, our lower bounds are linear in the classical input size. This realizes

[7] To see why, consider that on the diagonal of the truth table running top right to bottom left, we have that $x+y = 11...11$, the all 1's string. This means x and y must have non-zero values in non-overlapping locations, e.g. for two bits this diagonal consists of $(x = 00, y = 11)$, $(x = 01, y = 10)$, $(x = 10, y = 01)$ and $(x = 11, y = 00)$. Moving downward from any entry on that diagonal y becomes larger, so must now have an overlapping entry with the x string.

a long standing aspiration for f-routing and f-BB84 based position-verification schemes: that an honest party can compute a simple, classical function and perform $O(1)$ quantum operations, while a dishonest party must use entanglement that grows with the size of the classical input strings. However, this result comes at the unfortunate cost of requiring perfect correctness for the lower bounds to apply. Nonetheless, even proving bounds in this perfect case has so far been elusive, and we hope that our progress may open future avenues toward robust linear lower bounds.

An interesting observation is that our bounds lead to relations among operational quantities, e.g.

$$\mathrm{FR}_0(f) \geq \mathrm{QNP}^{cc}(f)$$

but are derived by separately relating FR_0 and QNP^{cc} to the non-deterministic rank. Given this result, it would be natural to expect a reduction at an operational level from f-routing to quantum non-deterministic communication complexity, but we know of no such reduction.

As a consequence of our lower bounds, we also obtain a new lower bound technique for randomness complexity in CDS with either perfect privacy or perfect correctness. We highlight the bound

$$\boxed{\mathrm{ppCDS}(f) \geq \Omega(\mathrm{QNP}^{\mathrm{cc}}(f)) \ .}$$

which appears to be new and incomparable to earlier bounds on ppCDS. It would be interesting to construct explicit functions for which this lower bound is tighter than earlier ones, for instance, bounds from the $\mathrm{PP}^{\mathrm{cc}}$ cost of f [4].

References

1 Rene Allerstorfer, Andreas Bluhm, Harry Buhrman, Matthias Christandl, Llorenç Escolà-Farràs, Florian Speelman, and Philip Verduyn Lunel. Making existing quantum position verification protocols secure against arbitrary transmission loss. *arXiv preprint*, 2023. arXiv:2312.12614.

2 Rene Allerstorfer, Harry Buhrman, Alex May, Florian Speelman, and Philip Verduyn Lunel. Relating non-local quantum computation to information theoretic cryptography. *arXiv preprint*, 2023. arXiv:2306.16462.

3 Harriet Apel, Toby Cubitt, Patrick Hayden, Tamara Kohler, and David Pérez-García. Security of position-based quantum cryptography limits hamiltonian simulation via holography. *arXiv preprint*, 2024. arXiv:2401.09058.

4 Benny Applebaum and Barak Arkis. On the power of amortization in secret sharing: d-uniform secret sharing and cds with constant information rate. *ACM Transactions on Computation Theory (TOCT)*, 2020. doi:10.1145/3417756.

5 Benny Applebaum and Prashant Nalini Vasudevan. Placing conditional disclosure of secrets in the communication complexity universe. *Journal of Cryptology*, 2021. doi:10.1007/s00145-021-09376-1.

6 Vahid R. Asadi, Richard Cleve, Eric Culf, and Alex May. Linear gate bounds against natural functions for position-verification. *arXiv preprint*, 2024. arXiv:2402.18648.

7 Vahid R Asadi, Kohdai Kuroiwa, Debbie Leung, Alex May, Sabrina Pasterski, and Chris Waddell. Conditional disclosure of secrets with quantum resources. *arXiv preprint*, 2024. arXiv:2404.14491.

8 László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory. In *27th Annual Symposium on Foundations of Computer Science (SFCS 1986)*. IEEE, 1986. doi:10.1109/SFCS.1986.15.

9 Salman Beigi and Robert König. Simplified instantaneous non-local quantum computation with applications to position-based cryptography. *New Journal of Physics*, 2011. doi:10.1088/1367-2630/13/9/093036.

10 Andreas Bluhm, Matthias Christandl, and Florian Speelman. A single-qubit position verification protocol that is secure against multi-qubit attacks. *Nature Physics*, 2022. doi:10.1038/s41567-022-01577-0.

11 Harry Buhrman, Nishanth Chandran, Serge Fehr, Ran Gelles, Vipul Goyal, Rafail Ostrovsky, and Christian Schaffner. Position-based quantum cryptography: Impossibility and constructions. *SIAM Journal on Computing*, 2014. doi:10.1137/130913687.

12 Harry Buhrman, Serge Fehr, Christian Schaffner, and Florian Speelman. The garden-hose model. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, 2013. doi:10.1145/2422436.2422455.

13 Joy Cree and Alex May. Code-routing: a new attack on position verification. *Quantum*, 2023. doi:10.22331/q-2023-08-09-1079.

14 Ronald de Wolf. Characterization of non-deterministic quantum query and quantum communication complexity. In *Proceedings of the 15th Annual IEEE Conference on Computational Complexity*, 2000. doi:10.1109/CCC.2000.856758.

15 Ronald De Wolf. Nondeterministic quantum query and communication complexities. *SIAM Journal on Computing*, 2003. doi:10.1137/S0097539702407345.

16 Llorenç Escolà-Farràs, Léo Colisson Palais, and Florian Speelman. A quantum cloning game with applications to quantum position verification. *arXiv preprint*, 2024. arXiv:2410.22157.

17 Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. *Journal of Computer and System Sciences*, 2000. doi:10.1006/jcss.1999.1689.

18 Adrian Kent, William J Munro, and Timothy P Spiller. Quantum tagging: Authenticating location via quantum information and relativistic signaling constraints. *Physical Review A*, 2011. doi:10.1103/PhysRevA.84.012326.

19 Adrian P Kent, William J Munro, Timothy P Spiller, and Raymond G Beausoleil. Tagging systems, 2006. US Patent 7,075,438.

20 Robert Malaney. The quantum car. *IEEE Wireless Communications Letters*, 2016. doi:10.1109/LWC.2016.2607740.

21 Alex May. Quantum tasks in holography. *Journal of High Energy Physics*, 2019. doi:10.1007/JHEP10(2019)233.

22 Alex May. Holographic quantum tasks with input and output regions. *Journal of High Energy Physics*, 2021. doi:10.1007/JHEP08(2021)055.

23 Alex May. Complexity and entanglement in non-local computation and holography. *Quantum*, 2022. doi:10.22331/q-2022-11-28-864.

24 Alex May, Geoff Penington, and Jonathan Sorce. Holographic scattering requires a connected entanglement wedge. *Journal of High Energy Physics*, 2020. doi:10.1007/JHEP08(2020)132.

25 Marco Tomamichel, Serge Fehr, Jędrzej Kaniewski, and Stephanie Wehner. A monogamy-of-entanglement game with applications to device-independent quantum cryptography. *New Journal of Physics*, 2013. doi:10.1088/1367-2630/15/10/103002.

26 Andreas Winter. Tight uniform continuity bounds for quantum entropies: conditional entropy, relative entropy distance and energy constraints. *Communications in Mathematical Physics*, 2016. doi:10.1007/s00220-016-2609-8.

A $\Omega(1)$ lower bounds on entanglement in f-routing

It is straightforward to establish an $\Omega(1)$ lower bound on entanglement for f-routing, though to our knowledge this lower bound is not recorded elsewhere in the literature, so we give it here. Note that for f-BB84 a $\Omega(1)$ lower bound follows from theorem 6.1 in [11].

To show the lower bound in the f-routing setting, we will first notice that in a f-routing task we can always replace a mixed resource state Ψ_{LR} with a pure state $|\Psi\rangle_{LR}$. Our lower bound will apply whenever $f(x, y)$ has an input $x = x_0$ such that $f(x_0, y) = F(y)$ is non-constant. We will assume perfect correctness of the f-routing protocol in our arguments but the generalization to the robust setting is immediate.

Considering the perfect protocol, we allow Alice and Bob to complete the easier task of performing f-routing when they know $x = x_0$. Alice, on the left, applies a quantum channel $\mathcal{N}_{QL\to AB}$, she keeps system A and sends B to Bob. Without decreasing their success probability we can take the isometric extension of this channel and have Alice keep the purifying system so that the state on ABR is pure. Bob, who knows y, knows where system Q should be brought so can, without decreasing their success probability, always simply forward the R system to whichever party should receive Q. Correctness of the protocol then requires that both AR and BR can recover Q.

Let Q be in the maximally entangled state $\Psi^{+}_{\bar{Q}Q}$ with reference system $\bar{Q}$. Then correctness of the protocol implies that

$$\begin{aligned} I(\bar{Q} : AR) &= 2 \log d_R \ , \\ I(\bar{Q} : BR) &= 2 \log d_R \ . \end{aligned} \tag{7}$$

We claim this implies that $S(R) \geq \log d_R$. First notice that purity of $\bar{Q}ABR$ and the second line above implies $I(\bar{Q} : A) = 0$. Then we apply the inequality,

$$I(\bar{Q} : AR) \leq I(\bar{Q} : A) + 2S(R) \ ,$$

which can be proven from strong subadditivity, and using the first line of Equation (7) we are done. In the robust setting, we can prove a similar bound by applying the above argument along with the continuity of the von Neumann entropy.

Stable Matching with Interviews

Itai Ashlagi
Stanford University, CA, USA

Jiale Chen
Stanford University, CA, USA

Mohammad Roghani
Stanford University, CA, USA

Amin Saberi
Stanford University, CA, USA

Abstract

In several two-sided markets, including labor and dating, agents typically have limited information about their preferences prior to mutual interactions. This issue can result in matching frictions, as arising in the labor market for medical residencies, where high application rates are followed by a large number of interviews. Yet, the extensive literature on two-sided matching primarily focuses on models where agents know their preferences, leaving the interactions necessary for preference discovery largely overlooked. This paper studies this problem using an algorithmic approach, extending Gale-Shapley's deferred acceptance to this context.

Two algorithms are proposed. The first is an adaptive algorithm that expands upon Gale-Shapley's deferred acceptance by incorporating interviews between applicants and positions. Similar to deferred acceptance, one side sequentially proposes to the other. However, the order of proposals is carefully chosen to ensure an interim stable matching is found. Furthermore, with high probability, the number of interviews conducted by each applicant or position is limited to $O(\log^2 n)$.

In many seasonal markets, interactions occur more simultaneously, consisting of an initial interview phase followed by a clearing stage. We present a non-adaptive algorithm for generating a single stage set of in tiered random markets. The algorithm finds an interim stable matching in such markets while assigning no more than $O(\log^3 n)$ interviews to each applicant or position.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory

Keywords and phrases Stable Matching, Gale–Shapley Algorithm, Algorithmic Game Theory

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.12

Funding The authors acknowledge the support of NSF awards 2209520, 2312156, and a gift from CISCO.

1 Introduction

In many two-sided matching markets used for dating or allocating labor, agents often interact prior to forming matches in order to gain information about their potential partners. These interactions, meant for learning preferences, can generate congestion, resulting in market inefficiencies. For example, in the market for fellowships in sports medicine, on average more than 23 applicants interview for each slot [33]. Similarly, in dating markets, such as the popular platform Tinder, only a small fraction of online conversations culminate in an in-person meeting [20, 15].

This paper considers the problem of how to alleviate interview congestion and improve welfare in two-sided matching markets. We take an algorithmic approach to form interviews and find stable matches while minimizing congestion.

We study this question in the classic two-sided marriage problem by [14]. In the model, there are n applicants and n positions. Each applicant a_i and position p_j has a publicly known value of u_i and v_j, respectively. The subjective interest of a_i for matching with p_j,


© Itai Ashlagi, Jiale Chen, Mohammad Roghani, and Amin Saberi;
licensed under Creative Commons License CC-BY 4.0



denoted by ε^A_{ij} is unknown unless the parties meet each other. The observed utility of a_i in p_j is $v_j + \varepsilon^A_{ij}$ if a_i and p_j have met, and equal to v_j otherwise. The distributions of ε^A_{ij}s are known, they are all independent, and their expected value is zero. The subjective interest ε^P_{ji} and the observed utility of positions in the applicants are defined similarly.

A matching is interim-stable if (i) all the matched pairs have interviewed each other and (ii) there are no blocking pairs with respect to the observed utilities. One simple approach for finding an interim stable matching is to ask all pairs to interview each other and then find a stable matching with respect to the observed utilities. That involves n interviews per agent. On the other hand, even in the special case in which all the u_is and v_js are equal to each other, finding an interim-stable matching requires at least $\Omega(\log n)$ interviews per agent.

The main result of the paper is to show that the number of interviews needed to find an interim-stable matching is much closer to the above lower-bound than the upper bound. In fact, it is possible to find an interim stable matching by assigning only a polylogarithmic number of interviews to each applicant or position. This is shown for both adaptive and non-adaptive models.

1.1 Approach

Adaptive algorithm. We present an adaptive algorithm that finds an interim stable matching while requiring that each applicant and position engage in at most $O(\log^2 n)$ interviews. The algorithm is applicable to a general setting, where the publicly known values assigned to applicants and positions can vary arbitrarily, and the subjective interests are independent and identically distributed.

▶ **Theorem** (See Theorem 1). *There exists an **adaptive** algorithm that finds an interim stable matching between n positions and n applicants such that with high probability every applicant and position participates in at most $O(\log^2 n)$ interviews.*

The proposed algorithm expands upon deferred acceptance [14] by dividing the process into distinct interview and proposal stages. Specifically, in the applicant-optimal version, each applicant submits a proposal to the position offering the highest observed utility. The position will accept the "highest" proposal or assign an interview if the proposing applicant has the potential to create a blocking pair, taking into account potential changes in the observed value following the interview.

The main technical ingredient of the analysis is to show the probability that an applicant gets rejected by more than $O(\log^2 n)$ consecutive positions is very small. More precisely, with probability at least $1 - n^{-3}$, each applicant a_i is matched to position p_j for $j \leq i + O(\log^2 n)$.

Non-adaptive algorithm. In the adaptive model, the algorithm is allowed to determine the sequence of interviews based on the results of previous ones. However, in practice, many markets favor a non-adaptive approach in which the interview lists are predetermined beforehand to allow the parties to streamline scheduling and the rest of the logistics.

We present a non-adaptive algorithm for interim stable matching with a poly-logarithmic number of interviews per agent. The algorithm works in tiered markets in which applicants and positions are partitioned into multiple tiers. Two applicants a_i and $a_{i'}$ in the same tier have the same publicly known value, i.e., $u_i = u_{i'}$. On the other hand, if an applicant a_i is in a higher tier than $a_{i'}$, then it is always preferred independent of the interview outcomes, i.e., for all position p_j, $\Pr[u_i + \varepsilon^P_{ji} > u_{i'} + \varepsilon^P_{ji'}] = 1$.

▶ **Theorem** (See Theorem 19). *There exists a* ***non-adaptive*** *algorithm that, with high probability, finds an interim stable matching for* ***tiered markets*** *with no more than* $O(\log^3 n)$ *interviews for each applicant or position.*

The design of the non-adaptive algorithm incorporates two key technical components. First, we observe that in markets where all applicants and positions are in a single tier, requiring each applicant to interview a set of $O(\log^2 n)$ positions chosen uniformly at random combined with an applicant-optimal Gale-Shapley algorithm results in an interim stable matching.

On the other hand, when all positions are in the same tier while applicants are in individual tiers, random interviews prove to be ineffective. This is primarily due to positions exhibiting a preference for applicants in the corresponding higher tiers. Consequently, these higher-tier applicants are given higher priority in the matching process. The challenge arises for the lowest-tier applicant who, despite undergoing a poly-logarithmic number of random interviews, struggles to find the sole available position among the interviewed positions. To address this, a different approach is taken. By fixing the order of positions, interviews are assigned to each applicant with consecutive positions whose indices closely match the tier number of the applicant. This strategy ensures that applicants with similar tier numbers have a substantial overlap of interviews. Consequently, positions that are interviewed by higher-tier applicants will be matched to them and will not be available options for lower-tier applicants.

The non-adaptive algorithm combines the above two technical ingredients with a few other ideas. Specifically, it creates a sequence of bipartite graphs $G_i(A_i, P_i, E_i)$, where each G_i may be a random graph like the first special case we considered, a sequential graph like the second special case, or a small complete bipartite graph when we have two small tiers. We refer the reader to Section 3.3 for the details of the algorithm and its analysis.

Further, in Section 3.4, we demonstrate that if applicants and positions have individual preferences within the tiers prior to the interviews, the exact same ideas and proofs presented in Section 3.3 carry over, with some mild assumptions about the distribution of individual preferences before the interviews and subjective interest arising from the interview.

The extensive literature on two-sided matching has led to a rich theory and successful designs in practice. However, most of the existing work focuses on models in which agents know their own preferences. Little is known about the interaction period in which agents learn about their preferences by interacting with each other. This paper attempts to address this gap by looking at this problem with an algorithmic lens, focusing on extending Gale-Shapley's deferred acceptance to this setting.

1.2 Related literature

Congestion and frictions in the early stages of two-sided markets stem from the lack of complete information and competition. Several approaches have been proposed to improve the screening process, including signaling, limiting the number of applications, and even coordinating interviews.

Signaling allows applicants to send programs "interest" signals [19, 10, 16], which intend to improve efficiency by helping in screening applications and interviewing applicants that are typically beyond reach. The literature has investigated how a limited number of signals indicating special interest can enhance efficiency. [16] demonstrate the impact of different signal quantities on the matching outcome. Signaling is used in the Economics job mar-

ket [9] and experimented in residency and fellowship markets [28, 30]. Instead of signaling mechanisms, this paper proposes interview mechanisms that address interview congestion by eliminating interviews that are unlikely to form a match.

Another approach considered is to limit the number of applications applicants can send. [1, 6, 32] demonstrate the benefit of limiting the number of applications when agents have complete information about their own preferences. We note that [6] and [32] further assume that agents are ex-ante symmetric.[1]

Several papers analyzed games induced by inviting agents for interviews [11, 17], demonstrating various frictions. Other CS papers study how to make interview decisions in the worst-case towards reaching a stable matching [12, 29].

A related approach for how to coordinate interviews is also studied. Specifically, [23] and [2] find that the match rate is high when agents have a similar number of interviews. [18] propose the idea of incorporating overlaps between positions. Our preference model allows for a more heterogeneous tiered market structure, which results in the non-adaptive algorithm assigning agents possibly an unequal number of interviews.

Our paper is inspired by [25, 24] who propose to conduct an interview match that will limit the number of interviews. Similar to our paper, [2] study non-adaptive interview mechanisms that generate many-to-many matching in large random markets with interim stability in the limit. In their model different applicants (positions) have heterogeneous pre-interview utilities of being matched to some position (applicant). Their heterogeneous utilities are a summation of common scores (generated from a continuous distribution) and idiosyncratic scores (and in particular the market is not tiered). Instead, we consider arbitrary homogeneous pre-interview utilities (i.e., just common scores) and propose an adaptive algorithm generating an interim stable matching for any finite market size. [2] further points to the challenge of handling tiered markets.[2] Instead our non-adaptive algorithm generates an interim-stable matching in every homogeneous tiered market.

This paper relies on the existing literature on random two-sided matching markets. Our non-adaptive algorithm incorporates the findings of [5] and [7] as a fundamental component. These prior studies specifically examine the average ranking of the matched positions for individual applicants in the applicant-proposing Gale-Shapley algorithm when preference lists are randomized. We further extend the tiered market model in [4] to allow for noisy preferences. [4] focuses on minimizing communication to reach an exact stable matching with high probability in a random market when agents know their own preferences. Instead, agents in our model know the public values and have perfect information only over public values and we seek to minimize the number of interviews required to learn about post-interview preferences in order to reach interim stability.

The concurrent work [3] extends our framework in two notable ways. First, it examines markets with *heterogeneous* pre-interview utilities, addressing scenarios where agents have differing pre-interview preferences. Second, it explores simple decentralized signaling mechanisms to determine which pairs should interview each other.

The authors demonstrate that in a single-tiered random market with sparse signals ($d = \omega(1)$), almost interim stability can be achieved. Specifically, the matching becomes perfectly interim stable after removing a vanishingly small fraction of agents. Furthermore, in the case of dense signals ($d = \Omega(\log^2 n/p)$), perfect interim stability can be attained in

[1] Limiting applications has also been proposed in for medical markets [31, 8, 26].

[2] They offer a heuristic for two-tiered markets; this heuristic adds to higher tier agents, safety interviews with lower tier agents.

imbalanced markets through short-side signaling. Additionally, they extend their results to multi-tiered markets and identify conditions under which signaling mechanisms are incentive compatible.

Finally, there is a growing literature in Economics that studies the structure stable matchings when agents have partial knowledge of their own preferences Liu et al. [22], Liu [21]. This literature does not consider the matching dynamics that arise prior to being matched. An exception is [13], which contributes to our understanding of the stages that occur prior to a match.

1.3 Our Model

Let $A = \{a_1, a_2, \ldots, a_n\}$ and $P = \{p_1, p_2, \ldots, p_n\}$ denote the set of applicants and positions respectively. The utility of applicant a_i for position p_j, v_{ij} can be written as $v_j + \varepsilon^A_{ij}$. The quantity v_j reflects the characteristics of the position, like working hours, salary, or prestige, and is public knowledge. Applicant a_i's subjective interest in position p_j is captured by ε^A_{ij}, which will only be revealed to the applicant after the interview. We assume ε^A_{ij}'s are independently sampled from a known symmetric distribution with mean zero. Define the utility of position j in applicant i, $u_{ij} = u_i + \varepsilon^P_{ji}$ similarly. Note that ε^A_{ij}'s and ε^P_{ji}'s are also independent from each other.

We define interim stable matchings. Let the *observed utility* of a_i in p_j, v^o_{ij} be $v_j + \varepsilon^A_{ij}$ if a_i and p_j have interviewed, and equal to v_j otherwise. Define u^o_{ji}'s or the observed utility of positions in the applicants similarly. A matching μ is interim stable if all the matched pairs have interviewed with each other and there are no blocking pairs with respect to observed utilities. In other words, $\{a_i, p_j\} \in \mu$ and $\{a_k, p_l\} \in \mu$ imply either $v^o_{ij} \geq v^o_{il}$ or $u^o_{ji} \geq u^o_{jk}$. Throughout the paper, we assume the applicants and positions prefer being matched to someone to remaining unmatched.

For ease of notation, we use $\mu(a_i)$ and $\mu(p_j)$ to refer to the position or the applicant they are matched to, respectively. Define $\mu(a_i) = \emptyset$ if a_i is not matched. We also use $\succ_a$ and $\succ_p$ to denote the preferences of applicants and positions, e.g., $p_j \succ_{a_i} p_{j'}$ if and only if $v^o_{ij} > v^o_{ij'}$.

We will study *adaptive* and *non-adaptive* algorithms for finding interim stable matching. Adaptive algorithms use the outcome of previous interviews to propose the next one. In contrast, non-adaptive algorithms determine the interviews that should be conducted between all the applicants and agents in one shot.

2 An Adaptive Algorithm

In this section, we present an adaptive algorithm for finding an interim stable matching. Our algorithm extends Gale-Shapley's deferred acceptance to incorporate interviews between the applicants and positions. Just like in deferred acceptance, one side makes proposals to the other sequentially, but the order of proposals is chosen carefully so that with high probability, the number of interviews needed to obtain interim stability is no more than $O(n \log^2 n)$.

▶ **Theorem 1.** *Algorithm 1 finds an interim stable matching between n positions and n applicants. Moreover, with high probability, the number of interviews done by each applicant or position is at most $O(\log^2 n)$.*

Let us start with an informal overview of the algorithm. Consider the applicant-proposing deferred acceptance mechanism. In the beginning, p_1 has the largest expected utility and is, therefore, the first choice for all applicants. Similarly, since a_1 is the applicant with the largest expected utility, it is p_1's first choice. The algorithm asks p_1 and a_1 to interview each

other. The interview may change a_1 and p_1's preferences. If they are still at the top of each other's preference list, we can match them and reject all other applicants for p_1; otherwise, we update the preference list of p_1 and a_1 and continue the process.

For an unmatched applicant a, let $\beta(a)$ be the most preferred position from which the applicant is not yet rejected. At any time, $\beta(a)$ is the position to which applicant a is going to propose next. At each step, we consider the position p_j with the smallest j such that there exists an applicant a_i where $\beta(a_i) = p_j$.

Suppose that among all proposals that p_j is receiving, a_i is the one that has the largest observed utility, i.e. $i = \arg\max_i u^o_{ij}$. If p_j and a_i have already interviewed each other, then the process is similar to deferred acceptance: if $\mu(p_j) \succ_{p_j} a_i$, position p_j rejects applicant a_i; otherwise, p_j rejects $\mu(p_j)$ and matches to a_i.

Now consider the case where p_j has not interviewed a_i. If $\mu(p_j) \succ_{p_j} a_i$ and $j < i$, p_j rejects a_i without an interview. Otherwise, p_j and a_i interview each other and update their observed utility and preference list. The algorithm ends when all agents are matched. See a formal description below.

Algorithm 1 Adaptive Algorithm for Interim Stable Matching.

Initialize $\mu(a) = \emptyset$ for $a \in A \cup P$, and $\forall i, j, v^o_{ij} = v_j, u^o_{ji} = u_i$.
while $\exists$ *an unmatched applicant* **do**
  Let j^* be the smallest index where $\beta(a_i) = p_{j^*}$, for some unmatched applicant a_i.
  Let a_{i^*} be position p_{j^*}'s favorite applicant from the set $\{a_i | \beta(a_i) = p_{j^*}, \mu(a_i) = \emptyset\}$.
  if (a_{i^*}, p_{j^*}) *have not interviewed and* $(i^* \le j^* \quad or \quad a_{i^*} \succ_{p_{j^*}} \mu(p_{j^*}))$ **then**
    Position p_{j^*} interviews applicant a_{i^*}; update $u^o_{i^*j^*}, v^o_{j^*i^*}$.
  else
    if $\mu(p_{j^*}) \succ_{p_{j^*}} a_{i^*}$ **then** p_{j^*} rejects applicant a_{i^*}.
    else
      p_{j^*} rejects applicant $\mu(p_{j^*})$ if it is not $\emptyset$.
      $\mu(a_{i^*}) \leftarrow p_{j^*}$ and $\mu(p_{j^*}) \leftarrow a_{i^*}$.
return matching μ.

2.1 Analysis of Algorithm 1

We will prove Theorem 1 in the rest of this section. We will show that in the course of the algorithm, all positions interview candidates that have a fairly similar global ranking. More specifically, with high probability, every position p_j only interviews applicants a_i where $j - O(\log^2 n) \le i \le j + O(\log n)$.

We start with a few simple observations about the algorithm. First, note that similar to the Gale-Shapley algorithm, when a position p_j gets matched to an applicant a_i, it remains matched until the end of the algorithm. Further, p_j may only reject a_i to match with a more preferable applicant.

▷ Claim 2. Suppose that unmatched position p_j and applicant a_i interview each other at some point during the algorithm and both observe non-negative ε^A_{ij} and ε^P_{ji}. Then, the algorithm (tentatively) matches them to each other. Subsequently, p_j may only interview applicants $a_{i'}$ for which $i' < \max(i, j+1)$.

Proof. If $\varepsilon_{ij}^A \geq 0$ and $\varepsilon_{ji}^P \geq 0$, the interview does not change $\beta(a_i)$ and applicant a_i remains the most preferable applicant proposing to p_j. Hence, the algorithm matches applicant a_i to position p_j in the next iteration.

Now, observe that, based on Algorithm 1 of Algorithm 1, p_j is going to interview an applicant $a_{i'}$ if either $i' \leq j$ or if $u_{i'} > u_i + \varepsilon_{ji}^P > u_i$, which implies $i' < i$. ◁

▶ **Observation 3.** *If position p_j interviews applicant a_i and $i < j$, then a_i is interviewed by all $p_{j'}$ where $i \leq j' < j$.*

Proof. Before proposing to p_j, a_i proposes to all positions $p_{j'}$ with $j' < j$. Also, based on Algorithm 1 of the algorithm, when $i \leq j'$, $p_{j'}$ does not reject a_i without an interview. ◀

The next lemma establishes that positions do not interview any applicant with a significantly higher index. Moreover, we show that with the possible exception of $8 \log n$ positions with the highest index (and lowest expected utility for the applicants), the positions get matched sequentially in the increasing order of their index.

▶ **Lemma 4.** *With probability of at least $1 - n^{-2}$, if p_j and a_i interview each other, then $i \leq j + 8 \log n$. Also, at the time an applicant a_i proposes to p_j,*

- *If $j < n - 8 \log n$, then $\mu(p_{j'}) \neq \emptyset$ for $j' < j$,*
- *If $j \geq n - 8 \log n$, then $\mu(p_{j'}) \neq \emptyset$ for $j' < n - 8 \log n$.*

Proof. The first part of the lemma holds trivially when $j \geq n - 8 \log n$. So suppose $j < n - 8 \log n$ and let S be the set of agents a_i with $i < j + 8 \log n$. At most $j - 1$ agents from S can be matched to position $p_{j'}$ for $j' < j$ at any time. Further, applicants propose to positions in the increasing order of their index. Hence, in order for p_j to get matched to an applicant $a_{i'}$ with $i' > j + 8 \log n$, it should reject at least $8 \log n$ applicants from set S. We will show that such an event is extremely unlikely.

Let $\mathcal{I}$ be the set of the indices of the first $8 \log n$ applicants interviewed by p_j. By the above argument, $\mathcal{I} \subseteq S$. By Claim 2, if p_j interviews some applicant a_i and observes that both ε_{ij}^A and ε_{ji}^P are non-negative, then p_j and a_i get matched. Since ε_{ij}^A and ε_{ji}^P are drawn independently at random from a symmetric distribution with mean zero, $\Pr[\varepsilon_{ij}^A \geq 0 \text{ and } \varepsilon_{ji}^P \geq 0] \geq 1/4$. Therefore,

$$\Pr[\nexists i \in \mathcal{I} \quad \varepsilon_{ij}^A \geq 0 \text{ and } \varepsilon_{ji}^P \geq 0] = \prod_{i \in \mathcal{I}} \Pr[\varepsilon_{ij}^A < 0 \text{ or } \varepsilon_{ji}^P < 0] \leq \left(\frac{3}{4}\right)^{8 \log n} \leq n^{-3}.$$

A union bound over the above events implies the probability that there exists an i such that a_i and p_j interview each other and $i > j + 8 \log n$ is at most n^{-2}.

The second part of the lemma follows by observing that as long as each position p_j with $j < n - 8 \log n$ gets matched to one of its first $8 \log n$ proposals, p_{j+1} does not receive any proposals before p_j is matched. ◀

For the rest of the section, we condition on the high probability events stated in Lemma 4. For $k \in [n]$, let $A_k = \{a_k, a_{k+1}, \ldots, a_n\}$ and $P_k = \{p_k, p_{k+1}, \ldots, p_n\}$. Also, let $\bar{A}_k = A \setminus A_k$ and $\bar{P}_k = P \setminus P_k$.

▷ Claim 5. For any $k \in [n]$,

$$\left|\{(a_i, p_j) \in \mu | i \geq k, j < k\}\right| \leq 8 \log n.$$

Proof. Since we condition on Lemma 4, $\{(a_i, p_j) \in \mu | i \geq k, j < k - 8\log n\} = \emptyset$. Therefore,

$$\left|\{(a_i, p_j) \in \mu | i \geq k, j < k\}\right| = \left|\{(a_i, p_j) \in \mu | i \geq k, k - 8\log n \leq j < k\}\right| \leq 8\log n. \quad \blacktriangleleft$$

▷ Claim 6. Suppose that the algorithm is considering the proposal between applicant a_{i^*} and position p_{j^*}. Then, for any $k \leq j^*$,

$$\left|\{(a_i, p_j) \in \mu | i < k, j \geq k\}\right| \leq 8\log n.$$

Proof. We need to prove the statement for $k < n - 8\log n$. By Lemma 4, when the algorithm is considering pair (a_{i^*}, p_{j^*}), we have $\mu(p_{j'}) \neq \emptyset$ for $j' < k$. By Claim 5,

$$\left|\{(a_i, p_j) \in \mu | i \geq k, j < k\}\right| \leq 8\log n$$

which implies,

$$\left|\{(a_i, p_j) \in \mu | i < k, j < k\}\right| \geq (k-1) - 8\log n. \tag{1}$$

On the other hand,

$$\begin{aligned}\left|\{(a_i, p_j) | i < k, j \geq k\}\right| &\leq (k-1) - \left|\{(a_i, p_j) | i < k, j < k\}\right| \\ &\leq 8\log n \qquad \text{(By } \textit{Equation } (1)\text{)}\end{aligned} \quad \triangleleft$$

▶ **Lemma 7.** *With a probability of at least $1 - 1/n$, if position p_j interviews applicant a_i then $j \leq i + 2000\log^2 n$.*

Proof. Let $\delta = 8\log n$ and $\gamma = 1999\log^2 n$. The statement holds for $i \geq n - \gamma - \delta$ since $n - i \leq \gamma + \delta < 2000\log^2 n$. Suppose for some $i < n - \gamma - \delta$, applicant a_i and position $p_{i+\gamma+1}$ interview each other at some point during the algorithm. At that time, by Lemma 4, all positions with index smaller than $i + \gamma + 1$ are matched to an applicant. Also, by Observation 3, all positions in $\{p_i, p_{i+1}, \ldots, p_{i+\gamma}\}$ have already interviewed a_i, but none of them is matched to a_i. We will show that the probability of such an event is at most n^{-4}.

Consider all positions p_l for $i \leq l \leq i + \gamma$ and define

$$S_l = \{a_i, a_{i+1}, \ldots, a_{l+\delta}\} \setminus \{\mu(p_{j'}) | i \leq j' < l\}.$$

By Claim 6, there are at most δ positions in the set of $\{p_{j'} | i \leq j' < l\}$ which are matched to an applicant with an index smaller than i. The rest are matched to an applicant in $\{a_i, a_{i+1}, \ldots, a_{l+\delta}\}$. Therefore, $|S_l| \leq 2\delta$.

Define Y_l to be indicating whether p_l's utility for matching to a_i is higher than being matched to all elements in S_l and X_l to be $Y_l = 1$, $\varepsilon^P_{li} \geq 0$, and $\varepsilon^A_{li} \geq 0$. By Claim 2, $X_l = 1$ implies that p_l can not be matched to an applicant with an index more than i. Further, by Claim 6 at most δ positions in $\{p_i, p_{i+1}, \ldots, p_{i+\gamma}\}$ may be matched to an applicant with index less than i, so it is sufficient to bound the probability that $X = \sum_{j=i}^{i+\gamma} X_j \leq \delta$.

Observe that $\Pr[Y_l = 1] \geq 1/(2\delta)$ for all l. Also, note that because of the independence of the subjective component of utilities, Y_l's are independent. Further, $E[X_l] = 1/4E[Y_l] \geq 1/(8\delta)$. Using Chernoff bound,

$$\Pr[X \leq \delta] \leq \Pr[|X - \mathbf{E}[X]| \geq \frac{\gamma}{16\delta}] \leq 2\exp\left(-\frac{\gamma^2/(16\delta)^2}{3\gamma/(8\delta)}\right) \leq 2\exp\left(-\frac{\gamma}{96\delta}\right) \leq n^{-2}.$$

The first inequality is due to $\mathbf{E}[X] - \frac{\gamma}{16\delta} \geq \frac{\gamma+1}{8\delta} - \frac{\gamma}{16\delta} > \frac{\gamma}{16\delta} \geq \delta$, assuming n is sufficiently large. Applying union bound over all applicants completes the proof. ◀

Proof of Theorem 1. The statements of Lemma 4 and Lemma 7 hold with high probability, for all applicants a_i. Therefore, every a_i is interviewed only by positions p_j where $j \in [\max(1, i - 8\log n), i + \min(n, 2000\log^2 n)]$. Similarly, for every position p_j in Algorithm 1, the position only interviews a_i's for which $i \in [\max(1, j - 2000\log^2 n), \min(n, j + 8\log n)]$. Therefore, the number of interviews done by each applicant or position is at most $O(\log^2 n)$.

The proof of stability is fairly similar to the analysis of deferred acceptance. Suppose there exists a blocking pair (a_i, p_j). This implies that $a_i \succ_{p_j} \mu(p_j)$ and $p_j \succ_{a_i} \mu(a_i)$. Therefore, a_i must have been rejected by p_j during one of the iterations of Algorithm 1. This implies that position p_j is matched with an applicant who has a higher observed utility, and the current match of p_j should not be worse than a_i, i.e., $\mu(p_j) \succ_{p_j} a_i$. That is a contradiction, as we initially assumed that (a_i, p_j) is a blocking pair. ◀

3 A Non-Adaptive Algorithm for Tiered Markets

The algorithm in the previous section forms the sequence of interviews adaptively, suggesting each interview based on the outcome of the earlier ones. However, many markets operate in a more simultaneous manner, where there is a stage of interviews followed by a clearing phase. Therefore, it may be more efficient if the interview lists are decided either in advance or independently for each position.

In this section, we present a non-adaptive algorithm for the problem in markets in which applicants and positions are partitioned into tiers, extending the model in [4] to allow for post interaction noise. Applicants have the same ex-ante utility for two positions in the same tier but prefer a position in a higher tier to a lower-tier position with probability 1. The same is true for positions.

More formally, let $0 = \tau_0 < \tau_1 < \ldots < \tau_M = m$ denote the tiers of positions. Remember the utility of an agent a_i for position p_j, $v_{ij} = v_j + \varepsilon^A_{ij}$. We will assume that $v_j = v_k$ if and only if p_j and p_k are in the same tier, i.e., $j, k \in [\tau_l + 1, \tau_{l+1}]$ for some l, and $v_j >> v_k$ if p_j is in a higher tier than p_k, i.e. $j \leq \tau_l < k$ for some l. Further, ε^A_{ij}'s are all independent and identically distributed, and their distributions are bounded and symmetric around 0. Therefore, $\Pr[v_{ij} \geq v_{ik}] = 1/2$ if p_j and p_k belong to the same tier and equal to 1 otherwise. We use *relative index* of a position to show the index of the position within the tier it belongs to. Formally, for position p_j where $\tau_{l-1} < j \leq \tau_l$, the relative index is equal to $j - \tau_{l-1}$. Define the tiers $0 = \gamma_0 < \gamma_1 < \ldots < \gamma_N = n$ for applicants similarly. We assume that the number of applicants and positions are equal in the general tiered markets, i.e. $n = m$, and every applicant and position will be matched. However, to develop the algorithm for the general tiered market, we use two subroutines that might need to solve a subproblem with $n \neq m$ case (see Section 3.1 and Section 3.2).

Our non-adaptive algorithm works in two phases. In the first phase, the algorithm proposes a set of interviews between the two sides. We represent this by a bipartite graph $G(A, P, E)$, where each edge $e \in E$ denotes an interview between its endpoints. In the second phase, after the interviews are conducted and the corresponding $\varepsilon^A_{ij}, \varepsilon^P_{ji}$ for $(a_i, p_j) \in E$ are revealed, the algorithm finds an interim stable matching between two sides. The main result of this section is that the proposed interviews in the first phase are such that (i) no agent or position has more than $O(\log^3 n)$ interviews, and (ii) after the interviews are done, the algorithm can find an interim stable matching supported by $G(A, P, E)$ with high probability.

In order to explain the main ideas of the algorithm, it will be instructive to look at two special cases in the following subsections.

3.1 Example 1: Single Tier Structure

In the first example, the applicants and positions are each in their own single tier, i.e. $N = M = 1$. The number of applicants and positions can be different[3] but assume that n and m are sufficiently large, and without loss of generality, $n \leq m$. In this case, first form G by connecting each applicant to δ positions chosen uniformly at random from the $P' = \{p_1, p_2, \ldots, p_n\}$.

Each applicant a_i interviews with δ positions and the distribution of ε_{ij}^A's are symmetrically distributed around 0. So, for every i, $\delta/2$ of the ε_{ij}^A's are going to be non-negative in expectation. In fact, a simple Chernoff bound shows that with high probability, for every applicant a_i, at least $\delta/3$ of positions p_j interviewing the applicant have a non-negative ε_{ij}^A.

▷ Claim 8. Among δ positions p_j that applicant a_i is interviewing, at least $\delta/3$ of them have $\varepsilon_{ij}^A \geq 0$ with probability $1 - \exp(-\delta/36)$.

Let π_i^A be the preference list of applicant a_i in the decreasing value of ε_{ij}^A (break ties randomly). It is not hard to see that each π_i^A is a random permutation chosen over the ordering of positions. Following the same argument, we can form π_j^P to be the random permutations representing the preference list of position p_j according to the decreasing values of ε_{ji}^P. The above observation allows us to take advantage of the following lemma.

▶ **Lemma 9** ([27], Theorem 6.1)**.** *Suppose that we run the applicant-proposing Gale-Shapley algorithm for the above random permutations. With probability $1 - O(m^{-c_t})$, each applicant a_i gets matched to a position with a rank less than $(2+t)\log^2 m$ in permutation π_i^A, where $c_t = 2t[3 + (4t+9)^{1/2}]^{-1}$.*

The above lemma combined with Claim 8 implies that every applicant a_i is matched with a position p_j where $\varepsilon_{ij}^A > 0$.

▶ **Lemma 10.** *For any t and δ such that $\delta \geq 3(2+t)\log^2 m$, with probability $1 - \exp(-\delta/36) - O(m^{-c_t})$, each applicant a_i gets matched to a position p_j that has interviewed a_i and $\varepsilon_{ij}^A > 0$, where $c_t = 2t[3 + (4t+9)^{1/2}]^{-1}$.*

3.2 Example 2: One Large Tier vs Multiple Singleton Tiers

In this example, all positions belong to the same tier, but each applicant is in a tier of its own. In other words, the applicants are ex-ante indifferent among positions, but all positions prefer applicant a_i to a_j when $i < j$.

Before presenting our solution, it may be worthwhile to understand why forming the interview graph randomly in the same fashion as our previous example is not suitable for this variation of the problem. First, note that since agent a_1 is the first choice for all the positions, a_1 should be matched to its most preferred position in every interim stable matching. Removing applicant a_1 and that position, the same argument applies to applicant a_2, and so on.

Now suppose that each applicant interviews δ random positions similar to the algorithm in Section 3.1. Since we choose δ random positions for applicant a_1 to interview and then applicant a_1 chooses the most preferred position after the interviews, all positions have

[3] Note that in the setting of the general tiered market, we assume $n = m$, however, the single tier structure serves as a subroutine of the algorithm for the general tiered market and might need to solve a subproblem with $n \neq m$.

the same probability of being the most preferred position for applicant a_1 in this process. Therefore, the process is equivalent to the case that we choose a random position for applicant a_1 and remove both, then we choose a random position for a_2 and remove both, and so on. Suppose the algorithm successfully matches the first $n-1$ applicants to their positions. When only the last applicant a_n remains, it is crucial to show that the only remaining position should be one of the positions that a_n interviewed. Otherwise, the resulting matching is not interim stable. Since we are removing positions randomly, the probability that the last remaining position is among the δ position that a_n interviewed is δ/m. Therefore, we cannot find an interim stable matching supported within G if we want $\delta = poly(\log m)$.

Instead, for this example, our algorithm forms G by connecting every agent a_i to positions p_j such that $j \in [\max(1, i-\theta), \min(n, i+\delta)]$, where $\theta = \Theta(\log^3 m)$ and $\delta = \Theta(\log^2 m)$. Note that the degree of applicants with an index close to 1 or n may be smaller than the rest of the applicants.

In this construction, applicants whose tiers are close to each other interview for sets of positions that are similar. Therefore, there is a high correlation between the positions to which they get matched. As a result, with high probability, for every i, the set of positions that have interviewed a_i and are not matched to one of the higher-tier applicants is non-empty. Further, a_i can find a stable match in this set. We will prove this statement in more generality in the next section when analyzing the algorithm for the general case.

3.3 An Algorithm for General Tiered Markets

We are now ready to present our algorithm and its analysis for tiered markets in their full generality. As we said before, the algorithm works in two phases. In phase 1, Algorithm 2 constructs a bipartite interview graph $G(A, P, E)$ between applicants and positions. This is done possibly in several iterations. In each iteration, i, the algorithm adds an edge set E_i between two subsets $A_i \subseteq A$ and $P_i \subseteq P$ to E. Further, it identifies whether the applicants or positions are going to be the proposing side in this part of the graph. Note that an applicant or a position can be in multiple A_is or P_is.

In the second phase, Algorithm 3 conducts the interviews between all the pairs in E and updates the preferences of the two sides. Then, in each iteration i, it removes vertices that are matched in earlier iterations and implements either a position or applicant-proposing deferred acceptance algorithm between A_i and P_i. We discuss the details of this algorithm later in the section.

In practice, one should first implement Algorithm 2 to construct the interview graph and then Algorithm 3 to implement the interviews and find the stable matching. But it will be useful for the analysis to couple the two algorithms and consider them together after each iteration i.

Algorithm 2 considers applicants and positions starting from the top. Let X and Y be the set of top-tier applicants and positions. Label the sets X and Y in such a way that $|X| \leq |Y|$. We will consider three different cases: (1) If both $|X|$ and $|Y|$ are relatively small, i.e. $|X| \leq \delta$ and $|Y| \leq \delta$ for $\delta = \Theta(\log^2 n)$, we can afford to set up an interview for each pair between X and Y. (2) If $|X| \leq \delta$ and $|Y| > \delta$, we choose a set S of size δ comprised of vertices of Y with the lowest index and set up an interview between each pair in S and X. (3) If both $|X|$ and $|Y|$ are large than δ, we use the same approach as Section 3.1. Specifically, for each vertex in X, we choose δ random vertices from the first $|X|$ positions in Y to interview.

When both $|X|$ and $|Y|$ are larger than δ, we can remove both X and the first $|X|$ vertices of Y from A and P and move to the next iteration of the algorithm. The situation is more subtle when for a short tier X, we choose a set $S \in Y$ which has a size larger than $|X|$. In this case, in the same iteration of Algorithm 3, some of the vertices of S will remain unmatched.

A priori, and without knowing the outcome of the interviews between these two sets, we do not know which vertices of S are going to remain unmatched. Therefore, we will not remove any vertex from Y immediately. Because of this subtlety, we will have to keep track of the "effective cardinality" $e(X)$ and $e(Y)$. They are updated to be equal to the number of unmatched vertices in X and Y, respectively, at the same iteration in Algorithm 3.

The algorithm continues in the same fashion. At every step, the top non-empty tiers X and Y from each side are selected and named so that $e(X) \leq e(Y)$. We call X the *short side* and Y the *long side*.

Algorithm 2 The Algorithm for Constructing the Interview Graph.

Let $\delta = 36 \log^2 n$ and $\theta = 72 \log^3 n$.
Initialize $e(X) = |X|$ for all tiers X as the number of unmatched vertices in the tier.
Initialize $G(A, P, E)$ to be an empty graph, $D \leftarrow []$, and $k \leftarrow 1$.
while $A \neq \emptyset$ *and* $P \neq \emptyset$ **do**
    Consider the top non-empty tiers of A and P and label them X and Y s.t. $e(X) \leq e(Y)$.
    if $e(Y) \leq \delta$ **then**
        Let E_k be the set of all edges between X and Y. ▷ Case 1
        $e(Y) \leftarrow e(Y) - e(X)$.
    else if $e(X) \leq \delta$ **then**
        Let S be the set of $\delta + |Y| - e(Y)$ vertices in Y with the lowest index. ▷ Case 2
        Let E_k be the set of all edges between X and S.
        $e(Y) \leftarrow e(Y) - e(X)$.
    else
        Let S be the set of $e(X) + |Y| - e(Y)$ vertices in Y with the lowest index. ▷ Case 3
        Form E_k by connecting each $x \in X$ to $\delta + |Y| - e(Y)$ random vertices of S.
        Remove all vertices of S from Y and then $e(Y) \leftarrow e(Y) - e(X) = |Y|$.
    if X *is in applicant side* **then** $D(k) \leftarrow$ 'applicant proposing';
    else $D(k) \leftarrow$ 'position proposing';
    Remove all vertices of X and $e(X) \leftarrow 0$.
    if $e(Y) = 0$ **then** remove all vertices of Y;
    if $|Y| - e(Y) \geq \theta$ **then** remove the $|Y| - e(Y) - \theta$ vertices of Y with lowest indices;
    $k \leftarrow k + 1$.
return $G(A, P, E = \bigcup_{i<k} E_i), D, k$.

In the rest of this section, we prove the correctness of the algorithm and give an upper bound on the number of interviews done by every applicant or position. First, we show that at any time during the algorithm $e(X)$ is close to $|X|$.

▶ **Invariant 11.** *For a tier X, after every iteration during the course of Algorithm 2, $e(X) \leq |X| < e(X) + \theta$.*

Proof. We prove this by induction on the number of iterations. Initially, $|X| = e(X)$. Now assume that $e(X) \leq |X| < e(X) + \theta$ before the ith iteration. If X is the short side in iteration i, then all vertices of X will be removed after this iteration, and thus $|X| = e(X) = 0$. Also, if X is the long side and the algorithm is in case 3, then $|X| = e(X)$ by Algorithm 2 of Algorithm 2.

In all other cases, $e(X)$ decreases after the iteration. Further, by Algorithm 2, if $|X| - e(X) \geq \theta$, we remove vertices of X for which $|X| < e(X) + \theta$. ◀

Also, it is not hard to see that all applications and positions will be removed by the end of Algorithm 2.

▶ **Observation 12.** *After the last iteration of Algorithm 2, all applicants and positions are removed.*

Proof. Let $X_1, \ldots, X_{\tau_M}$ be all tiers of positions and $Y_1, \ldots, Y_{\tau_N}$ be all tiers of applicants. Note that $\sum_{i=1}^{\tau_M} e(X_i) = \sum_{i=1}^{\tau_N} e(Y_i)$ at all times during the execution of Algorithm 2 since in all three cases, the effective cardinality of the short side tier is subtracted from both $\sum_{i=1}^{\tau_M} e(X_i)$ and $\sum_{i=1}^{\tau_N} e(Y_i)$. Since the algorithm does not terminate until either or both $\sum_{i=1}^{\tau_M} e(X_i) = 0$ or $\sum_{i=1}^{\tau_N} e(Y_i) = 0$, then all vertices are removed when the algorithm terminates. ◀

Algorithm 3 Algorithm for Finding a Stable Matching after the Interviews.

```
Let G(A, P, E), D, and k be the output of Algorithm 2.
Interview all edges of G.
for i = 1 to k − 1 do
    if D_i = 'applicant proposing' then
        Run applicant proposing Gale-Shapley on the unmatched endpoints of E_i.
    else
        Run position proposing Gale-Shapley on the unmatched endpoints of E_i.
return the matching.
```

We let A_i be the set of applicants that are endpoints of E_i. We define P_i similarly. Moreover, let $A'_i \subseteq A_i$ and $P'_i \subseteq P_i$ be the set of applicants and positions that are endpoints of E_i, and are unmatched when we run Gale-Shapley in Algorithm 3 between A_i and P_i. Therefore, if A_i and P_i belongs to tiers X and Y, then $|A'_i| = e(X)$ and $|P'_i| = e(Y)$ at the time of iteration i. As we mentioned before, a vertex may appear in multiple A_is or P_is. We will show that such sets are consecutive. To do so, we first give a helpful property of the algorithm.

▷ Claim 13. For a tier X that has interview with tiers $Y_1, Y_2, \ldots, Y_r$ where $e(Y_i) \leq \delta$ for all i, define X^{j-1} be tier X before interviewing with Y_j in Algorithm 2 for $j \in [1, r]$. Suppose $e(X^{j-1}) > \delta$ for $j \in [1, r]$, then all vertices in X^{j-1} with relative indices at most $S(X) + \delta - e(X^{j-1})$ will be selected for interview with Y_j, and $e(X^j) = e(X^{j-1}) - e(Y_j)$, where $S(X)$ is the initial size of tier X before the execution of Algorithm 2.

Proof. The algorithm always chooses vertices with the lowest index from the long side to interview. Consider the interview with Y_j, those are the vertices with a relative index of $S(X) - |X^{j-1}| + 1$ to $(S(X) - |X^{j-1}|) + (\delta + |X^{j-1}| - e(X^{j-1})) = S(Z) + \delta - e(X^{j-1})$. And by definition of Algorithm 2, $e(X^j) = e(X^{j-1}) - e(Y_j)$. ◁

▷ Claim 14. For each vertex v, there exist L and R such that v is in A_i or P_i iff $L \leq i \leq R$.

Proof. Let Z be the tier containing v and $S(Z)$ be the initial size of tier Z at the start of Algorithm 2. Also, let L be the first iteration considering v and $e(Z)$ be the effective size of tier Z at that point. It is not hard to see that if Z is the short side in this iteration, or if the short side Z' has size $e(Z') > \delta$, all the vertices of Z that have an interview in this iteration will be removed, and we are done. When Z is the long side and $e(Z) \leq \delta$, every vertex in Z will be interviewed.

The only remaining case to consider is where Z is on the long side, $e(Z) > \delta$, and the tiers considered alongside v are $Y_1, Y_2, \dots Y_r$, where $e(Y_i) \leq \delta$ for all i. From Claim 13 and the fact that $e(Z)$ decreases monotonically, as long as v is not removed from the graph, it will be selected for interview, which means it will be the endpoint of at least one edge in E_i. That implies v is in A_i or P_i by definition. ◁

▶ **Lemma 15.** *Suppose $a_r \in A'_i$, $|A'_i| \leq |P'_i|$, and $|P'_i| \geq \delta$. Then, with probability $1-O(1/n^2)$, a_r gets matched in iteration i of Algorithm 3 to some position $p_j \in P'_j$ such that $\varepsilon^A_{rj} > 0$.*

Proof. Let X and Y be two tiers that vertices of A_i and P_i belong to, respectively. Also, by the definition of A'_i, we have $|A'_i| = e(X)$ at the time of ith iteration. If $e(X) \leq \delta$, since $|P'_i| \geq \delta$, we choose a subset S of Y with size $\delta + |Y| - e(Y)$ and interview all edges. This set contains δ vertices that are not matched at ith iteration. By Lemma 10, if we choose $t = 10 \log^2 n / \log^2 e(Y)$, then $c_t > 2 \log n / \log e(Y)$ and with probability

$$1 - \exp(-\delta/36) - O\left(e(Y)^{-2 \log n / \log e(Y)}\right) \geq 1 - O\left(\frac{1}{n^2}\right),$$

a_r will match to a position $p_j \in P'_i$ such that $\varepsilon^A_{rj} > 0$.

Similarly, if $e(X) > \delta$, we choose a subset S of Y that contains $e(X)$ unmatched vertices. For each vertex of A'_i we interview δ random positions in P'_j. Hence, if we choose $t = 10 \log^2 n / \log^2 e(X)$ in Lemma 10, then $c_t > 2 \log n / \log e(X)$, and with probability

$$1 - \exp(-\delta/36) - O\left(e(X)^{-2 \log n / \log e(X)}\right) \geq 1 - O\left(\frac{1}{n^2}\right),$$

a_r will match to a position $p_j \in P'_i$ such that $\varepsilon^A_{rj} > 0$. ◀

▶ **Lemma 16.** *With probability $1-O(1/n^2)$, if a vertex is removed in iteration i of Algorithm 2, it will get matched via an edge in $\cup_{j \leq i} E_j$.*

Proof. Without loss of generality, assume that the removed vertex in iteration i is applicant a_r in tier X. Let Y be the tier of the other side that the algorithm considers in iteration i. First, assume that X is the short side, $e(X) \leq \delta$, and $e(Y) \leq \delta$ at iteration i. According to Case 1 of Algorithm 2, all vertices of X interview all vertices of Y, hence, the statement hold with probability 1. Now suppose that X is the short side and $e(Y) > \delta$ at iteration i. Hence, by definition, we have $|A'_i| \leq |P'_i|$ and $|P'_i| > \delta$. Therefore, by Lemma 15, a_r is matched to a position p_j such that $\varepsilon^A_{ij} > 0$ with probability $1 - O(1/n^2)$, which implies that $(a_r, p_j) \in E_i$. Furthermore, if $e(X) > \delta$, $e(Y) > \delta$, and X is the long side, $|A'_i| = |P'_i|$. Since for this case we run a position-proposing Gale-Shapley in Algorithm 3, by Lemma 15, all positions in P'_i get matched to applicants in A'_i with edges E_i. Thus, all vertices of A'_i are matched to positions in P'_i using edges E_i since we have $|A'_i| = |P'_i|$.

The only case that remains to be investigated is when X is the long side in several iterations until a_r is removed at iteration i, i.e. $a_r \in A_j$, $|A'_j| \geq |P'_j|$ for $j \in [l, i]$ and $|P'_j| \leq \delta$. Also, let $Y_l, Y_{l+1}, \dots, Y_i$ be the tiers of the other side in each iteration. Let $C_r = \sum_{l \leq j \leq i} e(Y_j)$. First, we prove that $C_r \geq \theta$.

Let X^{j-1} be tier X before the jth iteration of Algorithm 2 for $j \in [l, i+1]$. Also, let ψ be the relative position of a_r in tier X before any iteration. During iteration $j \in [l, i]$, since $|e(Y_j)| \leq \delta$, Claim 13 shows that (1) every vertex in X^j with index at most $S(X) + \delta - e(X^{j-1})$ is in A_j; (2) $e(X^j) = e(X^{j-1}) - e(Y_j)$. For X^{l-1}, the first case is that $e(X^{l-1}) = |X^{l-1}|$ and $S(X) - e(X^{l-1}) + 1 \leq \psi$ when there is no iteration on X before or when $|Y_{l-1}| > \delta$, and the second case is that $\psi > S(X) + \delta - e(X^{l-2}) \geq S(X) + e(Y_{l-1}) - e(X^{l-2}) \geq S(X) - e(X^{l-1})$

when $|Y_{l-1}| \leq \delta$. In both cases, we have $\psi \geq S(X) - e(X^{l-1}) + 1$. Since a_r is removed in iteration i, we have $S(X) - (e(X^i) + \theta) + 1 \geq \psi$, thus $C_r = \sum_{l \leq j \leq i} e(Y_j) = e(X^{l-1}) - e(X^i) \geq \theta$.

Now consider iteration j and suppose that a_r is not matched yet. Then, with a probability of at least $|P'_j|/\delta$, applicant a_r will be matched in iteration j because all applicants of A'_j are in the same tier, and subjective interests are i.i.d. Moreover, $|P'_j| = e(Y_j)$ by definition. Therefore, the probability that a_r remains unmatched after iteration i is at most

$$\prod_{j=l}^{i} \left(1 - \frac{|P'_j|}{\delta}\right) = \prod_{j=l}^{i} \left(1 - \frac{e(Y_j)}{\delta}\right) \leq \exp\left(-\frac{C_r}{\delta}\right) \leq \exp\left(-\frac{\theta}{\delta}\right),$$

which completes the proof because of our choice of δ and θ. ◀

Using a similar argument as Lemma 16, the next lemma gives a bound on the degree of every node in G.

▶ **Lemma 17.** *The number of interviews assigned to every position or applicant is at most* $O(\log^3 n)$

Proof. We will show the above for every applicant a_r. The proof for positions is the same. Let X be the tier that a_r belongs to. If X is the tier of the short side at iteration i, then a_r has at most $O(\theta)$ incident edges in E_i by definition of Algorithm 2 and Invariant 11. Also, if both $|A'_i| > \delta$ and $|P'_i| > \delta$, by Case 3 of the Algorithm 2 and Invariant 11, each vertex in A_i or P_i has at most $O(\theta)$ interviews in iteration i. Note that each vertex is in one of the above scenarios at most in one iteration since after that it gets removed. Therefore, similar to the proof of the previous lemma, it only remains to show that the number of interviews is bounded when X is the long side in several iterations until a_r is removed at iteration i, i.e. $a_r \in A_j$, $|A'_j| \geq |P'_j|$ for $j \in [l, i]$, and $|P'_j| \leq \delta$. Let $Y_l, Y_{l+1}, \ldots, Y_i$ be the tiers of the other side in each iteration.

Define C_r, ψ, and X^j for $j \in [l-1, i]$ as before. With a similar argument, we have $\psi \leq S(X) + \delta - e(X^{l-1})$, and

$$\begin{aligned}
\psi \geq S(X) - |X^{i-1}| + 1 &\geq S(X) - e(X^{i-1}) - (|X^{i-1}| - e(X^{i-1})) + 1 \\
&\geq S(X) - e(X^{i-1}) - \theta \\
&\geq S(X) - e(X^i) - (e(X^i) - e(X^{i-1})) - \theta \\
&\geq S(X) - e(X^i) - (\theta + \delta)
\end{aligned}$$

Thus, $C_r \leq e(X^{l-1}) - e(X^i) \leq \theta + 2\delta$. Note that the number of interviews for a_r is equal to $\sum_{i \leq j \leq i} |Y_j|$. For all tiers except Y_l, we have $e(Y_j) = |Y_j|$ at the time of the iteration j. Also, by Invariant 11, we have $|Y_l| \leq e(Y_l) + \theta$. Therefore,

$$\sum_{i \leq j \leq i} |Y_j| \leq e(Y_l) + \theta + \sum_{l < j \leq i} e(Y_j) = C_r + \theta \leq 2(\theta + \delta),$$

which implies the total number of interviews for vertex a_r is at most $O(\theta) = O(\log^3 n)$ before it gets removed which concludes the proof. ◀

▶ **Lemma 18.** *The matching produced by the algorithm is an interim stable matching with probability* $1 - O(1/n)$.

Proof. By Observation 12, all applicants and positions are removed from the graph after Algorithm 3 finishes. Using union bound for all vertices in Lemma 16, with probability $1 - O(1/n)$, all vertices are matched using an edge in E. Consider tiers X and X' from applicants and tiers Y and Y' from positions and suppose that X is a higher tier than X' and Y is higher than Y'. Let $H_1 = X \times Y'$ and $H_2 = X' \times Y$. We claim that at most one of the following holds: $H_1 \cap E \neq \emptyset$ or $H_2 \cap E \neq \emptyset$. Without loss of generality, suppose that $H_1 \cap E \neq \emptyset$. This implies that at some iteration i of the Algorithm 2, the algorithm considers tiers X and Y' in Algorithm 2. Hence, all vertices of Y should be already removed, which implies $H_2 \cap E = \emptyset$.

Now, consider applicant a_i and position p_j that are not matched to each other. We prove that (a_i, p_j) is not a blocking pair. Let $\mu(a_i)$ and $\mu(p_j)$ be the matches of a_i and p_j, respectively. Also, let X and Y be the tiers that include a_i and p_j, respectively. We use $X \succ X'$ to show tier X is preferable to tier X'. If $\mu(a_i) \in Y'$ such that $Y' \succ Y$, then this pair is not a blocking pair since $\mu(a_i) \succ_{a_i} p_j$. A similar argument works for $\mu(p_j)$. By the above argument, $\mu(a_i)$ and $\mu(p_j)$ does not belong to tiers Y' and X' such that $X \succ X'$ and $Y \succ Y'$. Hence, either or both $\mu(a_i) \in Y$ and $\mu(p_i) \in X$. Thus, X and Y are once considered at the same time in one iteration in Algorithm 2 of Algorithm 2.

Without loss of generality, suppose that at iteration r, the algorithm considers X and Y in Algorithm 2 and X is the short side. By definition, we have $a_i \in A'_r$. If $p_j \notin P'_r$, then $|P'_r| \geq \delta$, which implies that $\varepsilon^A_{i\mu(a_i)} > 0$. Thus, (a_i, p_j) is not a blocking pair. If $p_j \in P'_r$, since the algorithm finds a stable matching between A'_r and P'_r, then (a_i, p_j) is not a blocking pair, which finishes the proof. ◀

▶ **Theorem 19.** *There exists a non-adaptive algorithm that finds an interim stable matching for tiered ranking agents with high probability such that the number of interviews done by each applicant or position is at most* $O(\log^3 n)$.

Proof. By Lemma 18, the matching returned by the algorithm is interim stable. Furthermore, by Lemma 16, each applicant or position has at most $O(\log^3 n)$ interviews. ◀

3.4 Extension

In the aforementioned tiered market scenario, we presume the absence of ex-ante preferences within a tier. However, agents may have distinct individual preferences within tiers. For example, in the medical context preferences can be shaped by various factors, including geographical location, the availability of extracurricular activities, specific research laboratories they aspire to join, etc. To capture this characteristic of the market, we introduce the following model. Applicant a_i's utility v_{ij} for position p_j is now defined to be $v_{ij} = v_j + \eta^A_{ij} + \varepsilon^A_{ij}$, where $v_j = v_k$ if and only if p_j and p_k are in the same tier, and $v_j >> v_k$ if p_j is in a higher tier; η^A_{ij} is a small random perturbation of the ex-ante preference which is known before the interview; For now, we assume that η^A_{ij}s and ε^A_{ij}s are all drawn from the uniform distribution over [-1,1] independently. The same applies to positions.

We first briefly argue that our algorithm works for the extended tiered markets. Consider the first subproblem, Single Tier Structure.

- Using the same Chernoff bound arguments, we can prove that for each applicant, there is a constant c such that δ/c of the interviewed position has $\eta^A_{ij} + \varepsilon^A_{ij} > 1$.
- In the extended model, the preference lists are still random permutations over the randomness of both η and ε, thus each applicant will be matched to a position with a high rank.

Combine the above facts, we can give a similar argument as Lemma 10 that each applicant gets matched to an interviewed position with $\eta_{ij}^A + \varepsilon_{ij}^A > 1$. In this way, when we use it as a subroutine, it still satisfies the property that the applicant would never want to switch to positions that are not interviewed in the same tier since $\eta_{ik}^A < 1$.

Consider the second subproblem, One Large Tier v.s. Multiple Singleton Tiers. Using a similar Chernoff argument as above, it is guaranteed that a higher-tier applicant is matched to the most preferred available position that is interviewed with $\eta_{ij}^A + \varepsilon_{ij}^A > 1$. Thus they have no incentive to switch.

The proof for the extended general case is a combination of the two extended subproblems, and can be modified from the current one by changing the arguments in Lemma 15 from $\varepsilon_{ij}^A > 0$ to $\eta_{ij}^A + \varepsilon_{ij}^A > 1$.

▶ Remark. Although the boundedness of uniform distribution plays an important role in the analysis, more general distributions work, as long as with constant probability $\eta_{ij}^A + \varepsilon_{ij}^A$ is larger than the maximum of n i.i.d. η_{ik}^As.

4 Conclusion

The extensive body of work on two-sided matching has given rise to a comprehensive theory and efficient implementations in real-world scenarios. The majority of these studies assume that agents are fully aware of their preferences. This leaves a critical aspect – the interaction period where agents learn about their preferences via mutual interaction – largely under-explored. The current paper ventures into this area, using an algorithmic lens to extend Gale-Shapley's deferred acceptance to this setting.

We present an algorithmic approach that extends Gale-Shapley's deferred acceptance to include situations where agents form preferences during the interaction period. We propose two algorithms. The first, an adaptive algorithm, integrates interviews between applicants and positions into Gale-Shapley's deferred acceptance. Like deferred acceptance, one side sequentially proposes to the other. However, the order of proposals is arranged to increase the chances of achieving an interim stable match. Further, our analysis shows that the number of interviews conducted by each applicant or position is, with high probability, limited to $O(\log^2 n)$.

We also propose a non-adaptive algorithm for markets where the dynamics consist of an initial interview phase followed by a clearing stage. In these situations, having prearranged interview lists for each position may be necessary. Our non-adaptive algorithm creates these lists simultaneously. It aims to find a stable match in markets where applicants and positions are divided into tiers, and it limits the number of interviews for each applicant or position to no more than $O(\log^3 n)$.

We point out that the algorithms in this work do not account for the incentives of applicants or positions, i.e., mechanisms induced by the algorithm don't carry the dominant strategy incentive compatible (DSIC) property as the Gale-Shapley's algorithm.

References

1 Ishan Agarwal and Richard Cole. Stable matching: Choosing which proposals to make. *arXiv preprint*, 2022. doi:10.48550/arXiv.2204.04162.

2 Maxwell Allman and Itai Ashlagi. Interviewing matching in random markets, 2023. arXiv:2305.11350.

3 Maxwell Allman, Itai Ashlagi, Amin Saberi, and Sophie H. Yu. Stable matching with interviews, 2024.

4 Itai Ashlagi, Mark Braverman, Yash Kanoria, and Peng Shi. Clearing matching markets efficiently: informative signals and match recommendations. *Management Science*, 66(5):2163–2193, 2020. doi:10.1287/MNSC.2018.3265.

5 Itai Ashlagi, Yash Kanoria, and Jacob D Leshno. Unbalanced random matching markets: The stark effect of competition. *Journal of Political Economy*, 125(1):69–98, 2017.

6 Hedyeh Beyhaghi and Éva Tardos. Randomness and fairness in two-sided matching with limited interviews. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.

7 Linda Cai and Clayton Thomas. *The Short-Side Advantage in Random Matching Markets*, pages 257–267. SIAM, 2022. doi:10.1137/1.9781611977066.19.

8 J Bryan Carmody, Ilana S Rosman, and John C Carlson. Application fever: reviewing the causes, costs, and cures for residency application inflation. *Cureus*, 13(3), 2021.

9 Peter Coles, John Cawley, Phillip B Levine, Muriel Niederle, Alvin E Roth, and John J Siegfried. The job market for new economists: A market design perspective. *Journal of Economic Perspectives*, 24(4):187–206, 2010.

10 Peter Coles, Alexey Kushnir, and Muriel Niederle. Preference signaling in matching markets. *American Economic Journal: Microeconomics*, 5(2):99–134, 2013.

11 Joanna Drummond and Craig Boutilier. Elicitation and approximately stable matching with partial preferences. In *IJCAI*, pages 97–105. Citeseer, 2013. URL: http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6840.

12 Joanna Drummond and Craig Boutilier. Preference elicitation and interview minimization in stable matchings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28(1), 2014.

13 Federico Echenique, Ruy Gonzalez, Alistair J Wilson, and Leeat Yariv. Top of the batch: Interviews and the match. *American Economic Review: Insights*, 4(2):223–238, 2022.

14 David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

15 Trond Viggo Grøntvedt, Mons Bendixen, Ernst O Botnen, and Leif Edward Ottesen Kennair. Hook, line and sinker: Do tinder matches and meet ups lead to one-night stands? *Evolutionary Psychological Science*, 6(2):109–118, 2020.

16 Meena Jagadeesan and Alexander Wei. Varying the number of signals in matching markets. In *Web and Internet Economics: 14th International Conference, WINE 2018, Oxford, UK, December 15–17, 2018, Proceedings 14*, pages 232–245. Springer, 2018. doi:10.1007/978-3-030-04612-5_16.

17 Sangram V Kadam. Interviewing in matching markets with virtual interviews, 2021.

18 Robin S Lee and Michael Schwarz. Interviewing in two-sided matching markets. Working Paper 14922, National Bureau of Economic Research, April 2009. doi:10.3386/w14922.

19 Soohyung Lee and Muriel Niederle. Propose with a rose? signaling in internet dating markets. *Experimental Economics*, 18:731–755, 2015.

20 Leah E LeFebvre. Swiping me off my feet: Explicating relationship initiation on tinder. *Journal of Social and Personal Relationships*, 35(9):1205–1229, 2018.

21 Qingmin Liu. Stability and bayesian consistency in two-sided markets. *American Economic Review*, 110(8):2625–2666, 2020.

22 Qingmin Liu, George J Mailath, Andrew Postlewaite, and Larry Samuelson. Stable matching with incomplete information. *Econometrica*, 82(2):541–587, 2014.

23 Vikram Manjunath and Thayer Morrill. Interview hoarding, 2021. arXiv:2102.06440.

24 Marc L Melcher, Itai Ashlagi, and Irene Wapnir. Reducing the burden of fellowship interviews—reply. *JAMA*, 321(11):1107–1107, 2019.

25 Marc L Melcher, Irene Wapnir, and Itai Ashlagi. May the interview be with you: signal your preferences. *Journal of Graduate Medical Education*, 11(1):39–40, 2019.

26 Helen Kang Morgan, Abigail F Winkel, Taylor Standiford, Rodrigo Muñoz, Eric A Strand, David A Marzano, Tony Ogburn, Carol A Major, Susan Cox, and Maya M Hammoud. The case for capping residency interviews. *Journal of surgical education*, 78(3):755–762, 2021.

27 Boris Pittel. On likely solutions of a stable marriage problem. *The Annals of Applied Probability*, 2(2):358–401, 1992.

28 Steven D Pletcher, CW David Chang, Marc C Thorne, and Sonya Malekzadeh. The otolaryngology residency program preference signaling experience. *Academic Medicine*, 97(5):664, 2022.

29 Baharak Rastegari, Anne Condon, Nicole Immorlica, and Kevin Leyton-Brown. Two-sided matching with partial information. In *Proceedings of the fourteenth ACM conference on Electronic Commerce*, pages 733–750, 2013. doi:10.1145/2492002.2482607.

30 Parsa Salehi, Babak Azizzadeh, and Yan Lee. Preference signaling for competitive residency programs in the nrmp, December 2019. doi:10.4300/JGME-D-19-00695.1.

31 Aaron M Secrest, Garrett C Coman, J Michael Swink, and Keith L Duffy. Limiting residency applications to dermatology benefits nearly everyone. *The Journal of Clinical and Aesthetic Dermatology*, 14(7):30, 2021.

32 Erling Skancke. Welfare and strategic externalities in matching markets with interviews. *Available at SSRN 3960558*, 2021.

33 Ryan Sutton, William L. Wang, Walaa Abdelfadeel, Matthew Sherman, Lisa K. Cannada, and Chad A. Krueger. Are orthopedic fellowship programs giving out too many interviews? A retrospective analysis suggests they are. *HSS Journal®*, 19(2):210–216, 2023. doi:10.1177/15563316221103585.

A Deferred Proofs

▷ Claim 8. Among δ positions p_j that applicant a_i is interviewing, at least $\delta/3$ of them have $\varepsilon_{ij}^A \geq 0$ with probability $1 - \exp(-\delta/36)$.

Proof. Denote X_{ij} as the indicator random variable of whether $\varepsilon_{ij}^A \geq 0$, set P_i as the set of positions that applicant a_i is interviewing, and $X_i = \sum_{p_j \in P_i} X_{ij}$ as the total number of interviewed positions with $\varepsilon_{ij}^A \geq 0$. By definition, we know that X_{ij}'s are i.i.d. Bernoulli random variables with mean $1/2$. By Chernoff bound, we have

$$\Pr\left[X_i \leq \frac{\delta}{3}\right] \leq \Pr\left[X_i \leq \left(1 - \frac{1}{3}\right)\frac{\delta}{2}\right] \leq \exp\left(-\frac{\delta}{36}\right),$$

which finishes the proof. ◁

Low Sensitivity Hopsets

Vikrant Ashvinkumar
Rutgers University, Piscataway, NJ, USA

Aaron Bernstein
New York University, NY, USA

Chengyuan Deng
Rutgers University, Piscataway, NJ, USA

Jie Gao
Rutgers University, Piscataway, NJ, USA

Nicole Wein
University of Michigan, Ann Arbor, MI, USA

Abstract

Given a weighted graph $G = (V, E, w)$, a (β, ε)-hopset H is an edge set such that for any $s, t \in V$, where s can reach t in G, there is a path from s to t in $G \cup H$ which uses at most β hops whose length is in the range $[\mathsf{dist}_G(s,t), (1+\varepsilon)\mathsf{dist}_G(s,t)]$. We break away from the traditional question that asks for a hopset H that achieves small $|H|$ and small diameter β and instead study the *sensitivity* of H, a new quality measure. The sensitivity of a vertex (or edge) given a hopset H is, informally, the number of times a single hop in $G \cup H$ bypasses it; a bit more formally, assuming shortest paths in G are unique, it is the number of hopset edges $(s,t) \in H$ such that the vertex (or edge) is contained in the unique st-path in G having length exactly $\mathsf{dist}_G(s,t)$. The sensitivity associated with H is then the maximum sensitivity over all vertices (or edges). The highlights of our results are:

- A construction for $(\widetilde{O}(\sqrt{n}), 0)$-hopsets on undirected graphs with $O(\log n)$ sensitivity, complemented with a lower bound showing that $\widetilde{O}(\sqrt{n})$ is tight up to polylogarithmic factors for any construction with polylogarithmic sensitivity.
- A construction for $(n^{o(1)}, \varepsilon)$-hopsets on undirected graphs with $n^{o(1)}$ sensitivity for any $\varepsilon > 0$ that is at least inverse polylogarithmic, complemented with a lower bound on the tradeoff between β, ε, and the sensitivity.
- We define a notion of sensitivity for β-shortcut sets (which are the reachability analogues of hopsets) and give a construction for $\widetilde{O}(\sqrt{n})$-shortcut sets on *directed* graphs with $O(\log n)$ sensitivity, complemented with a lower bound showing that $\beta = \widetilde{\Omega}(n^{1/3})$ for any construction with polylogarithmic sensitivity.

We believe hopset sensitivity is a natural measure in and of itself, and could potentially find use in a diverse range of contexts. More concretely, the notion of hopset sensitivity is also directly motivated by the Differentially Private All Sets Range Queries problem [Deng et al. WADS 23]. Our result for $O(\log n)$ sensitivity $(\widetilde{O}(\sqrt{n}), 0)$-hopsets on undirected graphs immediately improves the current best-known upper bound on utility from $\widetilde{O}(n^{1/3})$ to $\widetilde{O}(n^{1/4})$ in the pure-DP setting, which is tight up to polylogarithmic factors.

2012 ACM Subject Classification Theory of computation → Shortest paths; Theory of computation → Sparsification and spanners

Keywords and phrases Hopsets, Shortcuts, Sensitivity, Differential Privacy

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.13

Related Version *Full Version*: `https://arxiv.org/abs/2407.10249` [4]

Funding *Aaron Bernstein*: Funding through Sloan Fellowship, Google Research Award, and NSF Grant 1942010.
Jie Gao: Ashvinkumar, Deng and Gao would like to acknowledge funding through NSF IIS-2229876, DMS-2220271, DMS-2311064, CCF-2208663, CCF-2118953.



© Vikrant Ashvinkumar, Aaron Bernstein, Chengyuan Deng, Jie Gao, and Nicole Wein;
licensed under Creative Commons License CC-BY 4.0



Nicole Wein: This work was done while the author was at DIMACS, Rutgers University, supported by a grant to DIMACS from the Simons Foundation (820931).

Acknowledgements The authors would like to thank Greg Bodwin for helpful discussions about hopsets.

1 Introduction

Many fundamental graph-theoretic problems involve the computation of reachability and shortest path distances. In a variety of computation models (e.g. parallel, distributed, streaming, dynamic), the computation of shortest paths of most state-of-the-art algorithms scales with the *hop-diameter* [7, 12, 13, 24–26, 28, 29, 33–35]. The hop-diameter of a graph refers to the maximum hop-length over reachable pairs of vertices, where the hop-length from vertex s to vertex t is the minimum number of edges in a shortest st-path. Motivated by the goal of reducing hop-diameter, Thorup introduced the notion of a *shortcut set* [44], which is a set of edges H added to a given graph G such that: $G \cup H$ has the same reachability structure as G (i.e. for all vertices s and t, s can reach t in $G \cup H$ if and only if s can reach t in G) and $G \cup H$ has small hop-diameter. The shortcut set was later generalized to the *hopset* by Cohen [16] (also informally by [35, 41]) which preserves (weighted) shortest distances in addition to reachability. Formal definitions are given below.

▶ **Definition 1** (Shortcut Set). *Given a graph $G = (V, E)$, a* β-shortcut set *of G is a set of edges $H \subseteq V \times V$ such that: (1) Every edge $(s, t) \in H$ is in the transitive closure of G. (2) For every vertex pair (s, t) in the transitive closure of G, there is an st-path in $G \cup H$ with at most β edges.*

▶ **Definition 2** (Hopset). Given a weighted graph $G = (V, E, w)$, a (β, ε)-*hopset* of G is a set of weighted edges $H \subseteq V \times V$ such that: (1) Every edge $(s, t) \in H$ has a weight of $\mathsf{dist}_G(s, t)$, where $\mathsf{dist}_G(s, t)$ stands for the shortest distance between s, t in graph G. (2) For every vertex pair (s, t) in the transitive closure of G, there is an st-path P_{st} in $G \cup H$ with at most β edges, and the weight of P_{st} is at most $(1 + \varepsilon)\mathsf{dist}_G(s, t)$.

A $(\beta, 0)$-hopset is sometimes called an exact hopset, while (β, ε)-hopsets for $\varepsilon > 0$ are called approximate hopsets. For both shortcut sets and hopsets, a natural measure of their *cost* is their size (i.e. the number of edges added). There has been a rich literature studying the tradeoff between the size of a shortcut/hopset and the hop-diameter for both directed and undirected graphs [2, 6, 8, 11, 20, 21, 30, 31, 36–40, 46, 47].

A New Problem: Low-Sensitivity Hopsets

In some settings, measuring the cost of a hopset by its total number of edges provides information that is too coarse and not descriptive enough to capture the problem. In this work, we instead consider a notion of cost that is more local to individual vertices and edges. Specifically, we define the notion of the *sensitivity* of a hopset[1].

Informally, given a graph $G = (V, E)$ and a hopset H, the sensitivity of a vertex v is the number of hopset edges that bypass v. We say a hopset edge $(s, t) \in H$ bypasses v if v is on the unique[2] st-path in G having length exactly $\mathsf{dist}_G(s, t)$. The vertex sensitivity associated

[1] To avoid redundancy, we use hopset as a general term for shortcut/approximate/hopset unless specified otherwise.

[2] We assume shortest paths are unique now. Later we formally define the conditions for non-unique shortest paths.

with H is then the maximum sensitivity over all vertices. The edge sensitivity is defined in a similar way. We denote the hopset vertex/edge sensitivity by $\|\widehat{\mathsf{v}}\|_\infty$ and $\|\widehat{\mathsf{e}}\|_\infty$[3], and defer the formal definitions to Section 3.1.

We say a hopset H has low sensitivity if it has a vertex/edge sensitivity of polylog(n). We also define an analogous notion of a *low-sensitivity shortcut set*. Since shortcut sets pertain to reachability instead of shortest paths, and there could be many paths between a vertex pair s, t, it is not clear a priori which path should absorb the sensitivity of a shortcut edge from s to t. We allow the algorithm to specify the paths; that is, the input is a graph, and the output is a path between each pair of vertices as well as a shortcut set that has low sensitivity with respect to the chosen paths.

A closely related concept is the *support size* of a hopset, first studied in [22] to get recently improved path-reporting distance oracles (PRDOs); see [14] for the latest on PRDOs. The support size of a hopset H is, loosely speaking, the number of edges in G that are bypassed by edges from H. This can be seen as the ℓ_0 norm of a certain vector whereas, in contrast, hopset sensitivity is the ℓ_∞ norm of the same vector (as the notation $\|\widehat{\mathsf{e}}\|_\infty$ suggests[4]).

At any rate, we investigate tradeoffs between hopset sensitivity and hop-diameter in this work. We believe hopset sensitivity is a natural measure in and of itself, and could potentially find use in a diverse range of contexts. In fact, low-sensitivity hopsets have already been implicitly studied in several prior works [15, 18, 23]. These prior works come from the area of differential privacy where sensitivity is a crucial concept since it captures the effect of the perturbation of individual data points on the output of an algorithm. Our new low-sensitivity hopset constructions directly improve the bounds from [18] for the problem of *Differentially Private Range Queries*. More details on this concrete application, as well as other applications of a more speculative nature, can be found in Section 1.2.

1.1 Our Results

We study upper and lower bounds for low-sensitivity shortcut sets, exact hopsets, and approximate hopsets, on both undirected and directed graphs.

To give context to our results, we draw parallels to the different regimes for traditional hopsets. For hopsets with $O(n)$ edges, there are essentially three diameter regimes: (1) directed shortcut/hopsets as well as undirected exact hopsets all provably require polynomial hop-diameter, (2) *approximate* hopsets for undirected graphs require only $n^{o(1)}$ hop-diameter, and (3) undirected shortcut sets trivially achieve diameter 2. Roughly speaking, these diameter regimes also hold for the low-sensitivity counterparts.

Our results focus on low-sensitivity undirected exact hopsets and directed shortcut sets from regime 1, as well as low-sensitivity approximate undirected hopsets from regime 2. For all of these, we prove both upper and lower bounds.

An overview of upper and lower bound results of all settings is shown in Table 1. Observe that the edge sensitivity of a hopset is always no more than the vertex sensitivity (formally proved in Section 3.2). Thus, it is always more desirable to have upper bounds for vertex sensitivity and lower bounds for edge sensitivity. All of our results are of this form.

[3] The arc is meant to be evocative of bypassing hopset edges. When they bypass vertices, we draw the arc over the symbol v and when they bypass edges, we draw the arc over the symbol e.

[4] In fact, we also look at the ℓ_1 norm in the proofs of our lower bounds in Section 5.

Table 1 Our Results for sensitivity and hop-diameter for shortcut sets, exact hopsets, and $(1+\varepsilon)$ hopsets. Regarding the undirected approximate hopset, the upper bound has $\varepsilon > 0$ at least inverse polylogarithmic and, in the lower bound, k is any positive integer and Δ is any small positive constant.

		Shortcut Set	Hopset	$(1+\varepsilon)$ Hopset
Undirected	U. B.	$\beta = O(\log^2 n)$ $\|\widehat{\mathsf{v}}\|_\infty = O(\log n)$ ([23])	$\beta = O(\sqrt{n}\log n)$ $\|\widehat{\mathsf{v}}\|_\infty = O(\log n)$ (Details in full version [4])	$\beta = n^{o(1)}$ $\|\widehat{\mathsf{v}}\|_\infty = n^{o(1)}$ (Details in full version [4])
	L. B.	–	$\|\widehat{\mathsf{e}}\|_\infty \cdot \beta^2 = \Omega(n)$ (Section 5.1)	$\beta = O_k((1/\varepsilon)^k)$ $\Rightarrow \|\widehat{\mathsf{e}}\|_\infty = \Omega(n^{\frac{1}{2^k-1}-\Delta})$ (Details in full version [4])
Directed	U. B.	$\beta = O(\sqrt{n}\log^3 n)$ $\|\widehat{\mathsf{v}}\|_\infty = O(\log n)$ (Details in full version [4])	$\beta = O(\sqrt{n}\log n)$ $\|\widehat{\mathsf{v}}\|_\infty = O(\sqrt{n}\log n)$ ([15, 18])	$\beta = O(\sqrt{n}\log n)$ $\|\widehat{\mathsf{v}}\|_\infty = O(\sqrt{n}\log n)$ (Implied by directed exact hopset)
	L. B.	$\|\widehat{\mathsf{e}}\|_\infty \cdot \beta = \Omega(n^{1/3})$ (Section 5.1)	$\|\widehat{\mathsf{e}}\|_\infty \cdot \beta^2 = \Omega(n)$ (Implied by undirected hopset)	$\|\widehat{\mathsf{e}}\|_\infty \cdot \beta = \Omega(n^{1/3})$ (Implied by directed shortcut set)

Undirected Exact Hopsets

Our first result is for low-sensitivity exact hopsets on undirected graphs. We show it is possible to achieve hop-diameter $\widetilde{O}(\sqrt{n})$ and $O(\log n)$ vertex sensitivity.

▶ **Theorem 3** (Undirected Exact Hopset Upper Bound). *There exists an algorithm producing an $(O(\sqrt{n}\log n), 0)$-hopset with $\|\widehat{\mathsf{v}}\|_\infty = O(\log n)$ over undirected and directed acyclic graphs.*

We complement our upper bound with a lower bound showing that a hop-diameter of $\widetilde{O}(\sqrt{n})$ is *tight* up to polylogarithmic factors, for any hopset with polylogarithmic (vertex- or edge-) sensitivity.

▶ **Theorem 4** (Undirected Exact Hopset Lower Bound). *Any construction of $(\beta, 0)$-hopsets H must have $\|\widehat{\mathsf{e}}\|_\infty \cdot \beta^2 = \Omega(n)$ for a graph G on n vertices.*

Our lower bound exhibits a smooth trade-off between hop-diameter and sensitivity. One could ask whether such a trade-off exists for upper bounds as well. As we show (see Remark 34), such a trade-off would imply the non-existence of certain *perfect path* systems, which is a big open problem in network design. The existence of such perfect path systems would imply, for example, better lower bounds for distance preservers [17].

Directed Shortcut Sets

For directed shortcut sets, we prove an upper bound with similar guarantees to our upper bound for undirected exact hopsets.

▶ **Theorem 5** (Directed Shortcut Set Upper Bound). *There exists an algorithm producing an $O(\sqrt{n}\log^3 n)$-shortcut set with $\|\widehat{\mathsf{v}}\|_\infty = O(\log n)$ for directed graphs.*

We complement our upper bound with a lower bound.

▶ **Theorem 6** (Directed Shortcut Set Lower Bound). *Any construction of β-shortcut sets H must have $\|\widehat{\mathsf{e}}\|_\infty \cdot \beta = \Omega(n^{1/3})$ for a directed graph G on n vertices.*

Unlike our bounds for exact undirected hopsets, our results for directed shortcut sets are not tight. In particular, for polylogarithmic vertex-sensitivity, there is a gap between hop-diameter $\widetilde{O}(n^{1/3})$ and $\widetilde{O}(\sqrt{n})$, which we leave as an open problem.

Undirected Approximate Hopsets

For undirected approximate hopsets, we prove that one can achieve much better hop-diameter than the polynomial bounds for the previously mentioned problems. In particular, we prove bounds of $n^{o(1)}$ for both sensitivity and hop-diameter when $\varepsilon > 0$ is at least inverse polylogarithmic.

▶ **Theorem 7.** *There exists an algorithm which produces a $\left(O\left((k/\varepsilon)^k \log^2 n\right), \varepsilon\right)$-hopset H with $\|\widehat{\mathsf{v}}\|_\infty = O(kn^{1/k} \log^2 n)$ for undirected graphs. For any $\varepsilon > 0$ that is at least inverse polylogarithmic, setting $k = \Theta(\sqrt{\log n})$ gives a $(n^{o(1)}, \varepsilon)$-hopset with $\|\widehat{\mathsf{v}}\|_\infty = n^{o(1)}$.*

We complement our upper bound with a lower bound.

▶ **Theorem 8.** *Fix a positive integer k and parameter $\varepsilon > 1/n^{o(1)}$. Any construction of (β, ε)-hopsets H with $\beta = O\left(\left(\frac{1}{2^{k-2}(2k-1)\varepsilon}\right)^k\right)$ has $\|\widehat{\mathsf{e}}\|_\infty \geq n^{\frac{1}{2^k-1}-\Delta}$, $\Delta > 0$.*

This lower bound is an exact analogue of the best known lower bound on the size of approximate hopsets in [2] (we just divide their size bound by n to get our sensitivity bound). For $\varepsilon > 0$ exactly inverse polylogarithmic, this lower bound says that it is not possible to have sensitivity and hop-diameter be polylogarithmic simultaneously, so we can only hope to improve either the sensitivity or diameter under this regime. We leave open whether we can get sensitivity and diameter simultaneously polylogarithmic when ε is larger than polylogarithmic (for example, constant ε).

To briefly address the problems that we do not focus on (recall that we focus on undirected exact hopsets, directed shortcut sets, and approximate undirected hopsets): Low-sensitivity undirected shortcut sets are simple (but not quite as trivial as in the traditional setting) as noted by prior work and in Observation 22. For directed hopsets, the upper bounds implicit in prior work [15, 18] remain the best-known, and from the lower bounds side, our results for undirected exact hopsets and directed shortcut sets immediately carry over to exact and approximate directed hopsets, respectively.

1.2 Applications

Our motivation for studying low-sensitivity hopsets is two-fold. The more concrete motivation is that they are already implicitly used in the problem of differentially private range queries, and our new upper bound for low-sensitivity hopsets (Theorem 3) immediately implies improved results for this problem (details below).

More generally, we believe that sensitivity is a natural measure for evaluating hopsets, as it is one way of modeling the "robustness" of a hopset: low sensitivity means that changing a vertex/edge in the underlying graph G only affects a small number of shortcuts in H. This corresponds, for example, to the following real-world scenario. In computer networking the notion of an overlay network [43] considers logical links layered on top of a physical network. The logical links support functionality in the overlay protocol but they are actually implemented along a physical path in the underlying network. Overlay networks have been

widely adopted in practice (e.g. VPN, VoIP, content delivery, P2P services) due to benefits such as encapsulation, ease of deployment, and quality of service requirements. The design of an overlay network could benefit from the resilience of low sensitivity hopsets – changes in one vertex or one edge in the underlying network only affect relatively few overlay links.

1.2.1 Differentially Private Range Query

Low-sensitivity hopsets have (implicitly) found applications in differential privacy (DP) [19]; this is the problem setting with which we are primarily motivated by. On a high level, differential privacy protects sensitive information by introducing perturbation, such that the output stays immune to the presence or absence of any individual's data. More formally,

▶ **Definition 9** (Differential Privacy)**.** For databases Y and Y' that differ on one data entry and $\epsilon > 0, \delta \geq 0$, a randomized algorithm $\mathcal{M}$ is (ϵ, δ)-*differentially private*, if for any measurable set A in the range of possible outputs, it holds that: $\Pr[\mathcal{M}(Y) \in A] \leq e^{\epsilon} \Pr[\mathcal{M}(Y') \in A] + \delta$.

The algorithm $\mathcal{M}$ is called pure-DP if $\delta = 0$ and approximate-DP otherwise[5].

Low-sensitivity hopsets have a direct application to the *All Sets Range Queries (ASRQ)* problem in differential privacy [18]. The input to ASRQ is an undirected graph G with public topology and shortest paths, and each edge is associated with a private attribute (which can be distinct from its weight). The output of ASRQ is an $n \times n$ matrix M, where entry (u, v) contains the summation of edge attributes along a shortest path between the vertex pair u, v. The DP-ASRQ problem asks for a private mechanism to output a matrix M' with the DP guarantee, while minimizing the utility or additive error, i.e. ℓ_∞ of $M - M'$. The best-known upper bound of the additive error from prior work is $\widetilde{O}(n^{1/3})$ for the pure-DP setting and $\widetilde{O}(n^{1/4})$ for the approximate-DP setting. The latter is tight up to polylogarithmic factors [10]. As we outline next, our low-sensitivity exact hopsets imply an improvement of the $\widetilde{O}(n^{1/3})$ bound for the pure-DP setting, down to the tight bound of $\widetilde{O}(n^{1/4})$, matching the approximate-DP setting.

We are able to show the connection between low-sensitivity hopsets and the DP-ASRQ problem by the following theorem, which is shown implicitly in [18].

▶ **Theorem 10.** *If there exists a $(\beta, 0)$-hopset H with edge-sensitivity $\|\widehat{\mathsf{e}}\|_\infty$ for any given undirected graph, then for any $\epsilon > 0$, there exists an ϵ-DP algorithm for the ASRQ problem such that the additive error is at most $\widetilde{O}(\frac{1}{\epsilon} \cdot \sqrt{\|\widehat{\mathsf{e}}\|_\infty \cdot \beta})$ with high probability.*

Plugging into Theorem 10 our $(\sqrt{n} \log n, 0)$-hopset with $\|\widehat{\mathsf{v}}\|_\infty = O(\log n)$ (and thus $\|\widehat{\mathsf{e}}\|_\infty = O(\log n)$) from Theorem 3, we obtain the following result:

▶ **Theorem 11** (ϵ-DP upper bound)**.** *There exists an ϵ-DP algorithm for the ASRQ problem such that the additive error is at most $\widetilde{O}(\frac{n^{1/4}}{\epsilon})$ with high probability.*

We show the proof of Theorem 11 in the full version [4] of the paper. We remark that low-sensitivity hopsets are also implicitly used in another problem studied in differential privacy: All Pairs Shortest Distances. However a direct relation like Theorem 10 is not available. We further discuss this problem in the full version [4] of the paper.

[5] Note that we are using "epsilon" in two different ways in this paper, but we disambiguate the symbols: ϵ for the differential privacy parameter, and ε for the approximation parameter of hopsets.

1.3 Organization

We first provide, in Section 2, a high-level overview of the technical challenges and ideas in our results. Formal preliminaries are then given in Section 3. Next, we show an upper bound for undirected exact hopsets in Section 4, leaving the upper bound for directed exact hopsets to the full version [4] of the paper. Our upper bounds for undirected approximate hopsets can also be found in the full version [4] of the paper. In Section 5, we show lower bounds on the sensitivity-diameter tradeoff for shortcut sets and hopsets, leaving bounds in the approximate setting to the full version [4] of the paper. We finish by listing several open problems in Section 6.

2 Technical Overview

In this section we outline the key technical ideas in our constructions. A natural starting point for proving upper and lower bounds for low-sensitivity hopsets is to simply look at bounds for traditional hopsets. As it turns out, these ideas can sometimes be massaged into the low-sensitivity setting, but they do not always give the best sensitivity bounds, which necessitates the development of new techniques.

We begin by discussing a technique from prior work on low-sensitivity hopsets that illuminates a key difference between the traditional and low-sensitivity settings.

Techniques from Prior Work: Heavy-Light Decomposition

We will start with a simple example: shortcut sets for undirected graphs. For traditional shortcut sets, this is a trivial problem. Simply add a star centered at an arbitrary vertex v to get a shortcut set with $n-1$ edges and diameter only 2. If we consider the sensitivity of this construction, we quickly realize that v has sensitivity $n-1$. Thus, this trivial solution does not work in the low-sensitivity setting. However, as noted implicitly in prior work [23], there is quite a simple construction that does work.

The solution is to use a *heavy-light decomposition* of the tree of routing paths rooted at v. A heavy-light decomposition partitions the edges of a tree into "light" edges and "heavy" paths, where any root-to-leaf path has $O(\log n)$ light edges. This is useful because we can independently shortcut each of the (vertex-disjoint) heavy paths down to diameter and sensitivity $O(\log n)$ using standard methods (see e.g. Figure 1). This results in a shortcut set with vertex-sensitivity $O(\log n)$ and diameter $O(\log^2 n)$. See Section 3.3.2 for more details.

This solution suggests a more general technique for constructing low-sensitivity hopsets: whenever a traditional hopset contains a star, simply replace it by the above heavy-light decomposition shortcutting scheme. This technique works, for instance, to translate the folklore exact hopset into the low-sensitivity setting. The folklore hopset with $\widetilde{O}(n)$ edges and $\widetilde{O}(\sqrt{n})$ hop-diameter is the following: randomly sample a set S of $\widetilde{O}(\sqrt{n})$ vertices and add a hopset edge between all pairs of reachable vertices in S (where the edge weight is the distance between its endpoints). Viewing each vertex in S as the center of a star, we can apply the heavy-light decomposition transformation to achieve both vertex-sensitivity and hop-diameter $\widetilde{O}(\sqrt{n})$ [15, 18]. This remains the best-known result for exact and approximate low-sensitivity directed hopsets.

We remark that it would also be natural to try to adapt Kogan and Parter's recent $\widetilde{O}(n^{1/3})$-diameter directed shortcut set [38] on $\widetilde{O}(n)$ edges, to the low-sensitivity setting. In short, it is not clear how to do this. The first step of their construction picks $\widetilde{O}(n^{2/3})$ chains on $\widetilde{O}(n^{1/3})$ vertices each, and shortcuts each of them. Shortcutting a single chain could add sensitivity to $\Omega(n)$ vertices and edges (since it is a chain, not a path). So, shortcutting $\widetilde{O}(n^{2/3})$ chains could already incur vertex and edge sensitivity $\widetilde{O}(n^{2/3})$.

Our Main Technical Contribution

Although the known constructions for low-sensitivity hopsets use the strategy of starting with traditional hopsets and applying the above heavy-light decomposition transformation, it is not clear that this is the optimal strategy. In particular, the low-sensitivity setting permits us freedom not allowed in the traditional setting: we can add as many edges as we'd like, as long as the sensitivity of each individual vertex is bounded. Thus, we would like to take advantage of our ability to add many edges, while ensuring that not too many of our added edges have overlapping underlying routing paths. In light of this new goal, we examine low-sensitivity hopsets from scratch, developing a new technique that is conceptually very different from any known techniques for the traditional setting.

Our main new technique allows us to improve above exact undirected hopset of prior work [15, 18] from vertex sensitivity $\widetilde{O}(\sqrt{n})$ all the way down to $O(\log n)$. Our resulting hopset with $O(\log n)$ vertex-sensitivity and $\widetilde{O}(\sqrt{n})$ hop-diameter is tight up to polylogarithmic factors in both parameters.

This technique heavily relies on the assumption that the routing paths are chosen to be *consistent*; that is, each pair of overlapping paths overlaps at one single contiguous subpath. This assumption holds for shortest paths in undirected graphs and DAGs (with a consistent tie-breaking scheme), but not for shortest paths in general directed graphs. This is why our technique does not apply to exact or approximate directed hopsets. But it does apply to directed shortcut sets when we use the technique on the DAG of SCCs, in combination with other ingredients such as carefully choosing routing paths.

Now, we outline the technique. It is a greedy algorithm for processing routing paths in a carefully chosen order. Each time we process a routing path, we shortcut the path using a simple and standard method (see Figure 1) which incurs $O(\log n)$ sensitivity. We need to be careful because if we shortcut a path that overlaps with a previously shortcutted path, the vertices in the overlap incur double the sensitivity. For this reason, when we process a path we only shortcut the segments of a path that have not been previously shortcutted. We think of it this way: every time a path is shortcutted, it cast "shadows" on other overlapping paths, and we only shortcut the non-shadowed segments of each path.

Using this method, it is immediate from the standard method for shortcutting a path, that the vertex-sensitivity is only $O(\log n)$. However, the hop-diameter is in question. The hop-diameter can be determined by roughly the number of shadows cast onto a path, since each shadow chops the path into more segments, and so the goal is to bound the number of shadows cast onto each path. Since the routing paths are consistent, we know that when we process a path it only casts at most one new shadowed segment (which may be further segmented by already existing shadows) onto every other path. This helps, but we still need to choose the order to process the paths carefully. Even with the consistency property, processing the paths in the wrong order could lead to $\Omega(n)$ vertex-sensitivity.

To choose the processing order, we carefully define a potential function and choose the path with maximum potential. The potential of a path P is rather simple to define and fast to implement: the number of vertices on P that are not on any previously chosen path. In our analysis, we show that this potential function yields an algorithm with the desired hop-diameter of $\tilde{O}(\sqrt{n})$.

Techniques for Lower Bounds

For our lower bounds, we use modifications of constructions that have previously been applied to traditional hopsets as well as spanners, emulators, distance preservers, reachability preservers, and related problems. These graph constructions are layered graphs defined by

a collection of *perfect paths*, which are unique paths (or unique shortest paths, depending on the setting) that go from the first to last layer and are pairwise edge-disjoint. These graphs are useful for traditional hopsets because their properties imply that the addition of a single edge can only decrease the hop-length of one perfect path. This condition is useful for low-sensitivity hopsets too, but the situation is more nuanced. When we add a single edge to a traditional hopset it simply counts 1 towards our budget of edges, whereas in the low-sensitivity setting the total amount of sensitivity added depends on the "length" of the added edge. Thus, we need to take into account all possible edge lengths, and consider their contributions to both sensitivity and diameter.

As a separate issue, we are interested in edge sensitivity for lower bounds. For this reason we modify known perfect path constructions by replacing each vertex by an edge, which requires a slightly more careful analysis.

Our final construction begins with a known perfect path construction and optimizes the parameters for the low-sensitivity hopset setting (which results in different graph parameters than constructions for traditional hopsets). We obtain a lower bound showing that to achieve polylogarithmic vertex or edge-sensitivity for exact undirected hopsets, the hop-diameter must be $\widetilde{\Omega}(\sqrt{n})$, which is tight with our aforementioned upper bound. We also obtain lower bounds for low-sensitivity directed shortcut sets and low-sensitivity approximate undirected hopsets, both by again appealing to known constructions of layered graphs.

▶ Remark 12. Bodwin and Hoppenworth [11] are able to prove stronger lower bounds on the number of edges in a hopset by relaxing the requirement that perfect paths are disjoint. Unfortunately, their relaxation does not immediately seem to be useful for lower bounding the *sensitivity* of a hopset; see the full version [4] of the paper for more details.

Techniques for Approximate Undirected Hopsets

For traditional hopsets with $O(n)$ size, as noted in Section 1, there are essentially 3 diameter regimes: (1) directed shortcut/hopsets as well as undirected exact hopsets all require polynomial hop-diameter, (2) *approximate* hopsets for undirected graphs require only $n^{o(1)}$ hop-diameter, and (3) undirected shortcut sets trivially achieve diameter 2. In the low-sensitivity setting, we have mainly discussed the polynomial diameter regime so far, but it also makes sense to consider the $n^{o(1)}$-hop-diameter regime for low-sensitivity approximate undirected hopsets. To this end, we extend known results for traditional approximate undirected hopsets to the low-sensitivity setting, achieving both vertex-sensitivity and hop-diameter $n^{o(1)}$ for any $\varepsilon > 0$ that is at least inverse polylogarithmic.

In the traditional setting, Huang-Pettie and Elkin-Neiman [20, 31] showed that Thorup-Zwick emulators [45] are optimal hopsets. Examining this construction, it is not clear if it has low vertex and edge sensitivity. Instead, we begin with the similar construction of Thorup-Zwick spanners (also in [45]). Recall that emulators and spanners are both sparse graphs that approximately preserve distances in graphs, but a spanner is a subgraph while an emulator can have added edges. Thus, it makes sense to consider an emulator as a hopset, but not a spanner (since it has no added edges). However, we can use the heavy-light tree decomposition shortcutting on top of the Thorup-Zwick spanner to obtain a construction that is similar to the Thorup-Zwick emulator, but now it has low sensitivity. Then, we can leverage the hopset analysis tools of Huang-Pettie to show that our $n^{o(1)}$ vertex sensitivity hopset also has hop-diameter $n^{o(1)}$ for any $\varepsilon > 0$ that is at least inverse polylogarithmic.

3 Preliminaries

3.1 Notations and Definitions

With the aim of formally defining vertex and edge sensitivity, we first introduce a couple of notions.

▶ **Definition 13** (Routing Paths)**.** Given a graph $G = (V, E)$ we call $\mathcal{P}$ a set of *routing paths* for G if for all $s, t \in V$ such that s can reach t, there is exactly one path $P \in \mathcal{P}$ such that P is a path in G with starting point s and ending point t. If G is weighted, we stipulate that the paths in $\mathcal{P}$ must be shortest paths in G.

Routing paths are meant to model paths that are deemed critical. In the context of hopsets, the routing paths must thus be shortest paths, whereas in the context of shortcut sets, any path may be chosen as a routing path.

▶ **Definition 14** (Span)**.** Given a graph $G = (V, E)$ and a tuple of a set of routing paths and a shortcut/hopset $\mathcal{A}(G) = (\mathcal{P}, H)$, we define the *span* of an edge $e = (s, t) \in H$ to be the unique path in $\mathcal{P}$ with starting point s and ending point t; we denote this with $\mathsf{span}_{\mathcal{A}(G)}(e)$. When the context is clear, we omit the subscript and write $\mathsf{span}(e)$.

We are now ready to define vertex and edge sensitivity. These are properties of (the inputs) graphs, together with (the outputs) a set of routing paths and a shortcut/hopset.

▶ **Definition 15** (Vertex Sensitivity)**.** Given a graph $G = (V, E)$ and a tuple of a set of routing paths and a shortcut/hopset $\mathcal{A}(G) = (\mathcal{P}, H)$, the *vertex sensitivity* vector, indexed by V, is denoted with $\widehat{\mathsf{v}}_{G,\mathcal{A}(G)}$. When the context is clear, we omit any subset of the subscripts: $\widehat{\mathsf{v}}$.

The value of $\widehat{\mathsf{v}}$ at index v is

$$\widehat{\mathsf{v}}(v) = |\{f \in H : v \in V(\mathsf{span}(f))\}|.$$

The worst case *vertex sensitivity* (given $G, \mathcal{A}(G)$) is

$$\|\widehat{\mathsf{v}}\|_\infty = \max_{v \in V} \widehat{\mathsf{v}}(v).$$

▶ **Definition 16** (Edge Sensitivity)**.** Given a graph $G = (V, E)$ and a tuple of a set of routing paths and a shortcut/hopset $\mathcal{A}(G) = (\mathcal{P}, H)$, the *edge sensitivity* vector, indexed by E, is denoted with $\widehat{\mathsf{e}}_{G,\mathcal{A}(G)}$. When the context is clear, we omit any subset of the subscripts: $\widehat{\mathsf{e}}$.

The value of $\widehat{\mathsf{e}}$ at index e is

$$\widehat{\mathsf{e}}(e) = |\{f \in H : e \in E(\mathsf{span}(f))\}|.$$

The worst case *edge sensitivity* (given $G, \mathcal{A}(G)$) is

$$\|\widehat{\mathsf{e}}\|_\infty = \max_{e \in E} \widehat{\mathsf{e}}(e).$$

A reason behind our choice of using vector notation to describe sensitivity is as follows: We sometimes consider the total sensitivity $\|\widehat{\mathsf{e}}\|_1 = \sum_e \widehat{\mathsf{e}}(e)$ and even the sum of sensitivities over a set $T \subseteq E$ which we write with $\mathbb{1}_T \cdot \widehat{\mathsf{e}}$ where $\mathbb{1}_T$ is the indicator vector of T and "$\cdot$" denotes the usual inner product over vectors.

Next, we define the notion of *consistency*, which will prove to be a very useful property of carefully chosen routing paths in our upper bound constructions.

▶ **Definition 17** (Consistency (Modification from [17])). A pair of paths P, P' are said to be *consistent* if their intersection contains at most one connected component. A set of paths $\mathcal{P}$ is said to be *consistent* if every pair of paths $P, P' \in \mathcal{P}$ are consistent.

A shortest path tiebreaking function chooses for each pair of vertices $s, t \in V$, where s can reach t, exactly one shortest path from s to t. We say a shortest path tiebreaking function is *consistent* if its image is consistent.

Our lower bounds go through via constructions of graphs inspired by [2, 9]; in fact we directly use Theorems 5 and 6 from the first paper and Theorem 4.6 from the second paper for our constructions, but our proofs and the way we invoke the theorems differ. A common object that appears in both papers, and that we use, are layered graphs.

▶ **Definition 18** (Layered Graphs). A graph $G = (V, E)$ is *layered* if the vertex set can be partitioned into $V = V_1 \sqcup V_2 \sqcup \ldots \sqcup V_\ell$ such that every edge goes between two consecutive parts V_i and V_{i+1}. If G is directed, then edges are oriented towards the parts with larger indices.

3.2 Relationship Between Vertex and Edge Sensitivity

We discuss how different notions of sensitivity relate to each other. First, observe that for a fixed shorctut/hopset H, its $\|\widehat{\mathsf{v}}\|_\infty$ and $\|\widehat{\mathsf{e}}\|_\infty$ could differ drastically. For example, consider an $n/2$-tipped star with tips of length 2; any H comprising of all edges from the center c to the ends of all the tips would have $\widehat{\mathsf{v}}(c) = n/2$ but $\widehat{\mathsf{e}}(e) = 1$ for all edges e. More generally, the shortcut/hopset edges with endpoint v contribute one each to $\widehat{\mathsf{v}}(v)$ whereas their contribution to the edge sensitivity can be dispersed over the routing paths incident to v. At any rate, observe that the edge sensitivity being no more than the vertex sensitivity generalizes to any graph.

▶ **Observation 19.** $\|\widehat{\mathsf{e}}_{G,\mathcal{A}(G)}\|_\infty \leq \|\widehat{\mathsf{v}}_{G,\mathcal{A}(G)}\|_\infty$.

Proof. Consider the edge $e = (u, v)$ where $\widehat{\mathsf{e}}(e) = \|\widehat{\mathsf{e}}\|_\infty$. Observe in particular that $\widehat{\mathsf{e}}(e) \leq \widehat{\mathsf{v}}(u)$ since every edge in $\mathcal{A}(G)$ contributing to $\widehat{\mathsf{e}}(e)$ also contributes to $\widehat{\mathsf{v}}(u)$. We therefore conclude that $\|\widehat{\mathsf{e}}_{G,\mathcal{A}(G)}\|_\infty \leq \|\widehat{\mathsf{v}}_{G,\mathcal{A}(G)}\|_\infty$. ◀

It follows from Observation 19 that proving upper bounds for $\|\widehat{\mathsf{v}}\|_\infty$ gets us the same bound for $\|\widehat{\mathsf{e}}\|_\infty$ and, in the contrapositive, proving lower bounds for $\|\widehat{\mathsf{e}}\|_\infty$ gets us the same bound for $\|\widehat{\mathsf{v}}\|_\infty$. We thus strive to prove upper bounds on $\|\widehat{\mathsf{v}}\|_\infty$ and lower bounds on $\|\widehat{\mathsf{e}}\|_\infty$ throughout the paper; all results presented here are of this form.

It is also useful to note that $|H| \leq n\|\widehat{\mathsf{v}}\|_\infty$ which says that small sensitivity implies hopset sizes which are not much larger than linear (i.e. none of our results produce prohibitively large hopsets). This follows, informally, since every vertex "sees" at most $\|\widehat{\mathsf{v}}\|_\infty$ hopset edges (the ones contributing to its sensitivity) and each hopset edge is "seen" at least once. Similar arguments will be described more formally in our lower bound proofs.

3.3 Algorithmic Building Blocks

In this section we describe primitive subroutines used in our constructions of shortcut/hopsets. The first tool addresses how to "shorten" paths; the second, which can be viewed as a generalization of the first, addresses how to shorten trees.

3.3.1 Path Shortcutting

PATH-HOPSET is a well-known primitive which, when given a path, reduces hop-lengths of shortest paths to $O(\log n)$ while adding no more than $O(\log n)$ sensitivity to any vertex. Intuitively, PATH-HOPSET adds an edge between the two endpoints of the path, then partitions the path into two equally sized subpaths and recurses into each of them. An example of the output of PATH-HOPSET is depicted in Figure 1.

Path-Hopset
Input: A path P from v_0 to v_ℓ
1. If $\ell < 2$ then return $\{\}$
2. Return $\{(v_0, v_\ell)\} \cup$ PATH-HOPSET$(P[v_0..v_{\lfloor \ell/2 \rfloor}]) \cup$ PATH-HOPSET$(P[v_{\lfloor \ell/2 \rfloor}..v_\ell])$

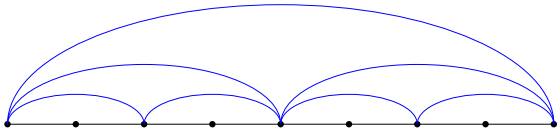


Figure 1 An example of PATH-HOPSET on a path of length 8 with vertices $v_0, \ldots, v_8$. The collection of blue edges is the shortcut/hopset output.

▶ **Observation 20.** *Given a path $P = v_0, v_1, \ldots, v_n$ (possibly oriented in the (v_i, v_{i+1}) direction), PATH-HOPSET outputs a $O(\log n)$-shortcut/hopset with $\|\hat{\mathrm{v}}\|_\infty = O(\log n)$.*

For completeness, we provide a proof in the full version [4] of the paper.

3.3.2 Tree Shortcutting via the Heavy-light Decomposition

TREE-HOPSET, described in [23], reduces the problem of shortcutting trees to that of shortcutting paths, where the instances passed to PATH-HOPSET are found via a heavy-light decomposition of the tree. A heavy-light decomposition [27] (also termed heavy path decomposition) of a rooted tree is a partitioning of its edges into "heavy" paths and "light" edges such that any root-to-leaf path passes through at most $O(\log n)$ light edges, which is captured by Proposition 21.

▶ **Proposition 21** ([27]). *Given a tree T of size n, there is a heavy-light decomposition such that*

- *any root-to-leaf path has at most $O(\log n)$ light edges;*
- *all heavy paths are vertex disjoint.*

Using the above proposition, TREE-HOPSET is specified as follows.

Tree-Hopset
Input: A tree T rooted at v
1. Return PATH-HOPSET(P) for all heavy paths P in a heavy-light decomposition of T

▶ **Observation 22.** *Given a tree T rooted at v (with possibly all edges oriented away from the root or all edges oriented towards the root),* Tree-Hopset *outputs an $O(\log^2 n)$-shortcut/hopset with $\|\widehat{\mathrm{v}}\|_\infty = O(\log n)$.*

This result is implicitly shown in the analysis of [23], but we provide a proof (in a differential-privacy-free language) in the full version [4] of the paper for completeness. Elementary use of Tree-Hopset yields the following quick results. A $O(\log^2 n)$-shortcut set construction with $\|\widehat{\mathrm{v}}\|_\infty = O(\log n)$ on undirected graphs. A directed $(O(\sqrt{n}\log n), 0)$-hopset construction with $\|\widehat{\mathrm{v}}\|_\infty = O(\sqrt{n}\log n)$. See the full version [4] of the paper for details.

4 Upper Bounds via a New Greedy Approach

This section describes a new way to construct low-sensitivity shortcut/hopsets when we are able to choose consistent routing paths; this is always possible, for example, in undirected graphs and DAGs. Using this approach, we can quite immediately get Theorem 3. With some slight care, we apply this approach to shortcut sets on directed graphs and obtain Theorem 5.

4.1 The Greedy Construction

At a schematic-level, the construction is very simple to describe: for each routing path P, processed in *some order*, apply Path-Hopset to the *uncovered pieces* of P. The details are filled in as follows.

Finding an Order to Process Paths

The order for which we process the shortest paths is described via a potential function Φ on routing paths P, which counts the number of vertices in P that have not yet been contained in a path that has thus far been processed.

▶ **Definition 23.** Let $\mathcal{P}$ be the set of all routing paths, and let $\mathcal{P}' \subseteq \mathcal{P}$ be a subset of the set of all routing paths. We define, for all $P \in \mathcal{P}$, the potential

$$\Phi_{\mathcal{P}'}(P) = |\{v \in V(P) : v \notin V(P') \text{ for all paths } P' \in \mathcal{P}'\}|.$$

If it is clear from the context, we omit the subscript and write $\Phi(P)$.

The i^{th} routing path in the order is then the one which maximizes $\Phi_{\mathcal{P}^{(i-1)}}$, where $\mathcal{P}^{(i-1)}$ is the set of the first $i-1$ routing paths in the order (that is, the ones that have already been processed).

The following is an observation that we will make use of later; it says that the potential of a path P is weakly decreasing throughout the construction process, which can be seen by noting that if P is processed then it has 0 potential and, otherwise, its vertices that are not contained in already processed paths can only drop as more and more paths are processed.

▶ **Observation 24.** *$\Phi_{\mathcal{P}^{(i)}}$ is weakly decreasing in i.*

Processing a Path

The uncovered pieces of an unprocessed path P depend on the the history of the algorithm up to the point where P is processed. These are the maximal subpaths of P for which no vertex is part of an already processed path.

▶ **Definition 25** (Uncovered pieces of a shortest path)**.** Let $\mathcal{P}'$ be the set of paths processed over so far. The *uncovered pieces* of P, given $\mathcal{P}'$, are then the maximal connected components of $\{v \in V(P) : v \notin V(P') \text{ for all paths } P' \in \mathcal{P}'\}$.

A path P is processed by running Path-Hopset on each of its uncovered pieces.

Putting Things Together

The algorithm producing an $\widetilde{O}(\sqrt{n}, 0)$-hopset with $\|\widehat{\mathrm{v}}\|_\infty = O(\log n)$ is then described by Greedy-Hopset which selects the path that maximizes the potential Φ and processes it as above, repeating until every path is processed.

Greedy-Hopset
Input: A graph $G = (V, E, w)$ with a consistent set of routing paths $\mathcal{P}$.
1. $\mathcal{P}' \leftarrow \emptyset, H \leftarrow \emptyset$
2. While $|\mathcal{P}'| < \binom{n}{2}$
   a. Select the next routing path $P^* \leftarrow \arg\max_{P \in \mathcal{P} \setminus \mathcal{P}'} \Phi_{\mathcal{P}'}(P)$, breaking ties arbitrarily
   b. $H \leftarrow H \cup$ Path-Hopset(p) for all uncovered pieces p of P^*
   c. $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{P^*\}$
3. Return H

One can quickly observe that Greedy-Hopset produces a low-sensitivity shortcut/hopset.

▶ **Observation 26.** *The hopset produced by Greedy-Hopset has* $\|\widehat{\mathrm{v}}\|_\infty = O(\log n)$.

Proof. Let v be an arbitrary vertex. Its sensitivity is only ever increased in one iteration of Greedy-Hopset, namely the first time a path P is chosen such that $v \in V(P)$. We know from Observation 20 that the contribution to $\widehat{\mathrm{v}}(v)$ in one iteration is bounded by $O(\log n)$, completing the proof. ◀

Before we turn to the remaining claim that Greedy-Hopset is an $\widetilde{O}(\sqrt{n}, 0)$-hopset, we need a couple of definitions. Henceforth, let the paths be processed in the order $P_1, P_2, \ldots, P_{O(n^2)}$. First we define the *shadows* and *penumbras* cast onto P_i. Shadows are, loosely speaking, complementary to the uncovered pieces of P_i when it is processed; they serve two main purposes: (i) to bound the number of uncovered pieces when P_i is processed, and (ii) as shortcuts that can be used to traverse the covered pieces of P_i.

▶ **Definition 27** (Shadows, Penumbras)**.** For any $j < i$, the *shadows* cast by P_j onto P_i are the segments of P_j that coincide with P_i after removing all shadows cast before P_j onto P_i. Formally, they are the maximal subpaths of

$$P_i^{(j)} = \{v \in P_i \mid v \in P_j \text{ and } v \notin P_k \text{ for } k < j\},$$

defined for all $j < i$.

Let S be a shadow cast onto P_i. We call the edges in $E(S, P_i \setminus S)$ the *penumbras* of S cast onto P_i, where $E(S, P_i \setminus S)$ denotes the edges with one endpoint in S and the other in $P_i \setminus S$.

Observe that a single path P_j may cast $\Omega(j)$ shadows onto P_i. Nevertheless, if P_i has many shadows cast onto it, there must be almost as many paths which cast shadows onto P_i.

▶ **Observation 28.** *Suppose after iterating P_1 through to P_t, there are k shadows cast onto P_i for $i > t$. Then, at least $(k-1)/2$ distinct paths among P_1 through to P_t cast at least one shadow onto P_i.*

Proof. First, note that since there are k shadows cast onto P_i, there are at least $k-1$ edges in P_i which are penumbras. While the path P_j may cast many penumbras onto P_i for $j < i$, by the consistency of routing paths P_j casts at most 2 penumbra onto P_i that have not already been cast before P_j; these edges are namely the ones incident to boundary of the intersection between P_j and P_i (see Figure 2). We conclude that there are at least $(k-1)/2$ distinct paths among P_1 through to P_t that cast at least one shadow onto P_i. ◀

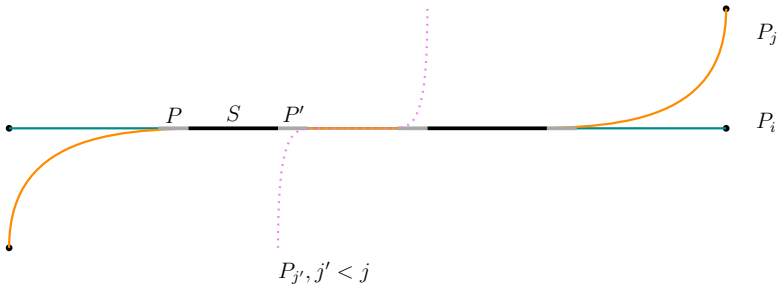


Figure 2 The penumbra P of S (a shadow cast by P_j onto P_i) is new, but the penumbra P' of S is also the penumbra of a shadow cast by $P_{j'}$ and is thus not new.

Suppose a routing path P from s to t has no more than k shadows cast onto it. Then the $G \cup \textsc{Greedy-Hopset}(G)$ hop-length from s to t is at most $\widetilde{O}(k)$; each shadow has shortcut-distance at most $O(\log n)$ by an application of Observation 20 to the time the shadow was first cast, and the remaining parts of P, of which there are $O(k)$ of them, each have shortcut-distance at most $O(\log n)$ by another application of Observation 20 at the time P is processed. Motivated by this, we now show that every routing path has at most $O(\sqrt{n})$ shadows cast onto it and every shadow on P has hop-length $O(\log n)$.

▶ **Lemma 29.** *P_i has $O(\sqrt{n})$ shadows cast onto it, for all i.*

Proof. Suppose, for a contradiction, that there is a routing path P_i which has at least $4\sqrt{n}$ shadows cast onto it. We aim to show that there are many "highly disjoint" paths that cast at least one shadow onto P_i, which will lead to a contradiction since too many such paths imply G has more than n vertices.

Towards this, let P_t for $t < i$ be the first path processed after which $\Phi(P_i) < \sqrt{n}$. Such a t must exist since, otherwise, we can invoke the monotonicity of Φ, which is weakly decreasing (see Observation 24), and Observation 28, which says that there must be at least $(4\sqrt{n}-1)/2$ distinct paths casting shadows onto P_i, to show that G has $\sqrt{n} \cdot (4\sqrt{n}-1)/2 > n$ vertices which is a contradiction.

After P_t is processed, there can be at most $\sqrt{n}$ more shadows cast onto P_i since $\Phi(P_i) < \sqrt{n}$ and each shadow cast onto P_i reduces $\Phi(P_i)$ by at least 1. There are therefore at least $3\sqrt{n}$ shadows cast onto P_i by paths processed before P_t (inclusive). Then, we can repeat the same argument as before: by Observation 24 and Observation 28 G has $\sqrt{n} \cdot (3\sqrt{n}-1)/2 > n$ vertices, leading to a contradiction. ◀

▶ **Observation 30.** *Let P be a routing path, and let S be any shadow with endpoints s, t cast onto P. Then the $G \cup$ Greedy-Hopset(G) hop-length from s to t (using edges contained in S or whose span is contained in S)[6] is bounded by $O(\log n)$.*

Proof. Let Q be the routing path casting S onto P. The claim goes through since at the time Q is processed, S is contained in an uncovered piece of Q; Greedy-Hopset makes a call to Path-Hopset on the uncovered piece of Q containing S when Q is being processed, finishing things up by Observation 20. ◀

4.2 An Upper Bound for Undirected Exact Hopsets

We are able to get upper bounds for exact hopsets on undirected graphs and DAGs immediately using Greedy-Hopset.

▶ **Theorem 3** (Undirected Exact Hopset Upper Bound)**.** *There exists an algorithm producing an $(O(\sqrt{n}\log n), 0)$-hopset with $\|\widehat{\mathbf{v}}\|_\infty = O(\log n)$ over undirected and directed acyclic graphs.*

Proof. We show the theorem for Greedy-Hopset. This follows from combining Lemma 29 and Observation 20, and also Observation 30. To spell things out, let G be an undirected or directed acyclic graph. Any routing path in G has $O(\sqrt{n})$ shadows, each shadow can be crossed within $O(\log n)$ hops (in the correct direction since G is undirected or a DAG), and the uncovered pieces of P at the time it is being processed (of which there are $O(\sqrt{n})$ many) can be crossed within $O(\log n)$ hops since Greedy-Hopset makes a call to Path-Hopset on each of these pieces.

The sensitivity claim follows directly from Observation 26. ◀

We leave our upper bound for directed exact hopsets, proving Theorem 5, to the full version [4] of the paper.

5 Lower Bounds: Sensitivity-Diameter Tradeoffs

In this section we show unconditional lower bounds for how low both the sensitivity and diameter of a construction can simultaneously be. Namely, we show Theorems 4 and 6 in Section 5.1 and leave the proof of Theorem 8 to the full version [4] of the paper.

5.1 Tradeoffs via Perfect Paths

Our lower bounds for exact hopsets and shortcut sets go through layered graphs endowed with a set of so-called perfect paths.

▶ **Definition 31** (Perfect Paths, Definition 8 in [9])**.** *Let $G = (V_1 \sqcup V_2 \sqcup \ldots \sqcup V_\ell, E)$ be a layered graph. A set of paths Π is* perfect *if each $\pi \in \Pi$ is the unique shortest path between its endpoints, each π starts in V_1 and ends in V_ℓ with exactly one node in each layer, and each $e \in E$ is in exactly one $\pi \in \Pi$.*

Layered graphs with perfect paths have been used to prove lower bounds for traditional shortcut/hopsets [17,30,32,37], and here we repurpose these constructions, with some changes, for our lower bounds. The "engine" of this section is Theorem 32 below, for which we first give some context before proving.

[6] This is an important constraint, since we wish for this claim to apply to $(\beta, 0)$-hopsets

▶ **Theorem 32.** *If there exists an ℓ-layered graph G with n nodes in each layer, and a set of perfect paths Π, then there exists a 2ℓ-layered graph G' with n nodes in each layer such that any ℓ-shortcut set or $(\ell, 0)$-hopset H on G' must have $\|\widehat{\mathsf{e}}\|_\infty \geq \frac{|\Pi|}{2n}$.*

Simplifying Assumptions

First, observe that if G is unweighted and we are concerned with directed ℓ-shortcut sets, then for any s that reaches t, the length of any st-path is completely determined by the layers they belong to; any shortcut set on G is thus also a hopset on G (where all edges of G are given weight 1). It is therefore sufficient to show the theorem for $(\ell, 0)$-hopsets.

Definition of the graph G'

We can show a lower bound for vertex sensitivity using G from the hypothesis of Theorem 32 directly (in fact, the proof is simpler). However, to get a lower bound for *edge* sensitivity, a little more care needs to be taken: we add a 0 weight edge in the "middle" of each vertex of G, which gives us G'. We now describe G' more formally.

G' is a standard transformation of G comprising of (i) vertices v^{in}, v^{out} for all $v \in V(G)$; (ii) 0 weight edges (v^{in}, v^{out}) for all $v \in V(G)$; (iii) edges (u^{out}, v^{in}) for all $(u, v) \in E(G)$, with the same weight. The proof will only look at the set of paths Π' in G' which are "lifted" from Π (dually, which project down back to Π), formally defined as follows:

$$\Pi' = \left\{(v_1^{in}, v_1^{out}, v_2^{in}, v_2^{out}, \ldots, v_\ell^{in}, v_\ell^{out}) : (v_1, v_2, \ldots, v_\ell) \in \Pi\right\}.$$

Observe that every path $\pi' \in \Pi'$ is the unique shortest path between its endpoints (which starts in the first layer and ends in layer 2ℓ), a property inherited from Π being a set of perfect paths. From this, notice that there is no choice in selecting the routing path from s to t for any $s, t \in \pi'$ for all $\pi' \in \Pi'$; any such routing path must be the subpath of π' from s to t since π' is the unique shortest path between its endpoints, and so the span of hopset edges (s, t) are fixed. Moreover, each edge of the form (u_{out}, v_{in}) is in exactly one $\pi' \in \Pi'$ since Π is perfect. Finally, note that $|\Pi'| = |\Pi|$. We are now ready to state the proof.

Proof of Theorem 32. Let G' and Π' be defined as above, and let H be any $(\ell, 0)$-hopset for G'. Let $H' \subseteq H$ be a minimal set of edges which ensures that there are ℓ hop-length shortest paths in $G' \cup H'$ between the endpoints of π' for all $\pi' \in \Pi'$. Observe that $\|\widehat{\mathsf{e}}_{G',H}\|_\infty \geq \|\widehat{\mathsf{e}}_{G',H'}\|_\infty$ since $H' \subseteq H$ and so it suffices to lower bound the latter which we now write without subscripts.

Let T be the set of all edges of the form (v_{in}, v_{out}) and $\mathbb{1}_T$ be its indicator vector; note for later that $|T| = \|\mathbb{1}_T\|_1 = n\ell$. We will lower bound the total sensitivity of edges in T and use an averaging argument to get a bound on $\|\widehat{\mathsf{e}}\|_\infty$. More specifically, we will show

$$\|\widehat{\mathsf{e}}\|_\infty \geq \mathbb{1}_T \cdot \widehat{\mathsf{e}} / \|\mathbb{1}_T\|_1 \geq x/2$$

where $\|\widehat{\mathsf{e}}\|_\infty \geq \widehat{\mathsf{e}} \cdot \mathbb{1}_T / \|\mathbb{1}_T\|_1$ comes from a simple averaging argument[7]. To this end, consider the process where we add the edges of $H' = \{e_1, e_2, \ldots, e_{|H'|}\}$ to G' one by one so that $H'_0 = \{\}$ and $H'_i = H'_{i-1} \cup \{e_i\}$, and define the potential

$$\Phi_i = \sum_{\pi' \in \Pi'} \left| E(P_{G' \cup H'_i}(\pi')) \right|$$

where $P_{G' \cup H'_i}(\pi')$ denotes a shortest path in $G' \cup H'_i$ with the smallest hop-length between the endpoints of π'. Φ_i is then the sum of said hop-lengths running over $\pi' \in \Pi'$ at stage i.

[7] To be overly indulgent, from an application of Hölder's inequality.

Observe that the initial value Φ_0 is $2|\Pi|\ell$ (since there are $|\Pi| = |\Pi'|$ paths in Π', each with hop-length exactly 2ℓ), and the final value $\Phi_{|H'|}$ is at most $|\Pi|\ell$ (since H' is an $(\ell, 0)$-hopset). Moreover, at most one summand of Φ_i differs from the corresponding summand of Φ_{i-1} for the following reason: $\mathsf{span}(e_i)$ is contained entirely and only in one $\pi' \in \Pi'$ by the minimality of H' and uniqueness of π'. For the same reason, $\mathsf{span}(e_i)$ alternates between edges not in T and edges in T and so twice the contribution of e_i to $\mathbb{1}_T \cdot \widehat{\mathsf{e}}$ is at least the decrease in potential $\Phi_{i-1} - \Phi_i$. That is,

$$\Phi_i + 2\mathbb{1}_T \cdot \widehat{\mathsf{e}}_{G',H'_i} > \Phi_{i-1} + 2\mathbb{1}_T \cdot \widehat{\mathsf{e}}_{G',H'_{i-1}}.$$

Using our observation about the initial and final values of Φ, we conclude that

$$\begin{aligned} & |\Pi|\ell + 2\mathbb{1}_T \cdot \widehat{\mathsf{e}} \geq \Phi_{|H'|} + 2\mathbb{1}_T \cdot \widehat{\mathsf{e}} > \Phi_0 + 2\mathbb{1}_T \cdot \widehat{\mathsf{e}}_{G',H'_0} = 2|\Pi|\ell \\ \implies & \mathbb{1}_T \cdot \widehat{\mathsf{e}} \geq \frac{|\Pi|\ell}{2} \\ \implies & \frac{\mathbb{1}_T \cdot \widehat{\mathsf{e}}}{\|\mathbb{1}_T\|_1} \geq \frac{|\Pi|}{2n}. && (\|\mathbb{1}_T\|_1 = n\ell) \end{aligned}$$

We conclude from the last line that $\|\widehat{\mathsf{e}}\|_\infty \geq \frac{|\Pi|}{2n}$. ◀

Finally, we can invoke Theorem 32 with known constructions to get unconditional lower bounds on the sensitivity-diameter tradeoff.

▶ **Proposition 33** (Theorem 5 of [9], earlier versions of which appear in [17, 42])**.** *For any integers $n, \ell \leq n$, and $x \leq n/\ell$, there is an ℓ-layered weighted graph G with n nodes in each layer and a set of perfect paths Π such that each node is in exactly x paths in Π.*

▶ **Theorem 4** (Undirected Exact Hopset Lower Bound)**.** *Any construction of $(\beta, 0)$-hopsets H must have $\|\widehat{\mathsf{e}}\|_\infty \cdot \beta^2 = \Omega(n)$ for a graph G on n vertices.*

Proof. For large enough N, we invoke Proposition 33, setting

- $n \leftarrow \frac{N}{2\beta}$
- $\ell \leftarrow 2\beta$
- $x \leftarrow \frac{N}{4\beta^2}$;

call this graph G_N. Note that $|\Pi|/(2n) = x/2$. We then pass each G_N into Theorem 32 to get

$$\|\widehat{\mathsf{e}}\|_\infty \geq \frac{x}{2} = \frac{N}{8\beta^2} \implies \|\widehat{\mathsf{e}}\|_\infty \cdot \beta^2 \geq \frac{N}{8},$$

which completes the proof. ◀

Observe that Theorem 4 shows that Theorem 3 is tight up to polylogarithmic factors, but it remains open whether we can explicitly trade between the sensitivity and diameter. For example, can we find a $(\widetilde{O}(n^{1/2-c}), 0)$-hopset with $\|\hat{\mathsf{v}}\|_\infty = \widetilde{O}(n^{2c})$ for any c? We close the discussion on exact hopsets with a remark on a connection between the problem of packing perfect paths in layered graphs and a possible threshold effect in the upper bounds for sensitivity-diameter tradeoffs.

▶ Remark 34. Observe that Theorem 32 connects the existence of layered graphs with many perfect paths to sensitivity-diameter tradeoffs, and this concretely manifests as a $\|\widehat{\mathsf{e}}\|_\infty \beta^2 = \Omega(N)$ bound by using the construction of Proposition 33. What are the ramifications of being able to pack even more perfect paths into an ℓ-layered graph? Let us explore one

particular hypothetical scenario. Suppose it is possible to pack N perfect paths into $\sqrt{N}$-layered graphs. Then, we are able to show (in exactly the same way as Theorem 4, but with different parameters) an $\|\hat{\mathsf{e}}\|_\infty \beta = \Omega(N)$ bound for $\beta \leq \sqrt{N}/2$. It is conceivable that such a construction may exist[8]. If such a packing does indeed exist for infinitely many values of N, this would rule out the possibility of achieving a smooth sensitivity-diameter tradeoff on the upper bound side. By Theorem 3, Greedy-Hopset produces a $(\widetilde{O}(\sqrt{n}), 0)$-hopset with $O(\log n)$ sensitivity, and any polynomial improvement to β would raise the sensitivity very abruptly from $O(\log n)$ to $\Omega(\sqrt{n})$. That is to say, a $(\widetilde{O}(n^{1/2-c}), 0)$-hopset construction could hope for no better than $\|\hat{\mathsf{v}}\|_\infty = \widetilde{\Omega}(n^{1/2+c})$ if we can pack N perfect paths into $\sqrt{N}$-layered graphs. On the flip side, this means that achieving a tradeoff like a $(\widetilde{O}(n^{1/2-c}), 0)$-hopset construction with $\|\hat{\mathsf{v}}\|_\infty$ less than $\widetilde{O}(n^{1/2+c})$ by a polynomial factor, for some constant $c > 0$, would rule out the existence of $\sqrt{N}$-layered graphs packed with N perfect paths. A similar line of reasoning also shows that any polynomial improvement to β (keeping $\|\hat{\mathsf{v}}\|_\infty = \widetilde{O}(\sqrt{n})$) over Folklore-Hopset will rule out the existence of $\sqrt{N}$-layered graphs packed with N perfect paths. The existence of layered graphs packed with as many perfect paths as possible is an open problem related to many lower bounds in the hopset/distance preservers/etc literature [11, 17]; particularly, we would get more streamlined lower bounds for hopsets matching the result of [11].

We next show a lower bound for reachability in directed graphs.

▶ **Proposition 35** (Theorem 6 of [9], earlier versions of which appear in [1, 3, 5, 17, 32]). *For any integers $n, d \geq 2, \ell \leq n^{1/d}$, and*

$$x = O\left(n^{\frac{d-1}{d+1}} \ell^{-d\frac{d-1}{d+1}}\right),$$

there is an ℓ-layered unweighted graph G with n nodes in each layer and a set of perfect paths Π such that each node is in exactly x paths in Π.

▶ **Theorem 6** (Directed Shortcut Set Lower Bound). *Any construction of β-shortcut sets H must have $\|\hat{\mathsf{e}}\|_\infty \cdot \beta = \Omega(n^{1/3})$ for a directed graph G on n vertices.*

Proof. For large enough N, we invoke Proposition 35, setting

- $n \leftarrow \frac{N}{2\beta}$
- $d \leftarrow 2$
- $\ell \leftarrow 2\beta$
- $x \leftarrow \Theta(\frac{N^{1/3}}{\beta})$;

call this graph G_N. Note that $|\Pi|/(2n) = x/2$. We then pass G_N into Theorem 32 to get

$$\|\hat{\mathsf{e}}\|_\infty \geq \frac{x}{2} = \Theta(\frac{N^{1/3}}{\beta}) \implies \|\hat{\mathsf{e}}\|_\infty \cdot \beta \geq \Omega(N^{1/3}),$$

which completes the proof. ◀

Curiously, none of our upper bounds and lower bounds for directed shortcut/hopsets match each other. New ideas are probably needed to better understand sensitivity-diameter tradeoffs (for both shortcut sets and hopsets) in the directed setting.

To see our lower bounds for approximate hopsets, proving Theorem 8, see the full version [4] of the paper.

[8] And such a construction does not contradict Theorem 3 since, there, $\beta \geq \sqrt{N} \log n > \sqrt{N}/2$.

6 Open Problems

In this section we collect a list of problems left open by our results. First and foremost, the picture pertaining to *directed* graphs is least clear.

- The bounds for shortcut sets (see Theorem 5 and Theorem 6) on directed graphs are not tight. Can we provide tight results for any specific value of sensitivity (say $O(\log n)$) if not for all values?
- The best known upper bound for directed exact hopsets is based on the folklore algorithm for traditional hopsets. It had been a long standing open problem whether this was the best we could do, and it is only within the last year that the folklore algorithm has been shown to be optimal [11]. Can similarly fresh ideas close the gap between the folklore algorithm and lower bounds here?
- We know essentially nothing insightful about the situation for directed approximate hopsets, besides the fact that our shortcut set lower bound in Theorem 6 applies (since a directed approximate hopset is a directed shortcut set). Find a non-trivial upper bound.

While more is known for undirected graphs, there are certainly gaps left to fill.

- The upper bound given by Greedy-Hopset only works for a specific sensitivity regime (i.e. it completes the picture for only one point on the tradeoff curve). Find a non-trivial algorithm, that outputs hopsets close to the curve of Theorem 4, where the sensitivity is tunable to values larger than polylogarithmic. See Remark 34 for more context surrounding this problem.
- Our lower bounds for undirected approximate hopsets in Theorem 8 are not tight; they do not address the regime when ε is a constant well. For a constant $\varepsilon > 0$, do there exist (β, ε)-hopsets with β and $\|\widehat{\mathrm{v}}\|_\infty$ simultaneously polylogarithmic, or must one of the quantities necessarily be superpolylogarithmic?

Finally, we believe that hopset sensitivity is a natural notion which should have concrete algorithmic applications beyond differential privacy. Where else can low-sensitivity hopsets can be used as a primitive?

References

1. Amir Abboud and Greg Bodwin. The 4/3 additive spanner exponent is tight. *Journal of the ACM (JACM)*, 64(4):1–20, 2017. doi:10.1145/3088511.
2. Amir Abboud, Greg Bodwin, and Seth Pettie. A hierarchy of lower bounds for sublinear additive spanners. *SIAM Journal on Computing*, 47(6):2203–2236, 2018. doi:10.1137/16M1105815.
3. Noga Alon. Testing subgraphs in large graphs. *Random Structures & Algorithms*, 21(3-4):359–370, 2002. doi:10.1002/RSA.10056.
4. Vikrant Ashvinkumar, Aaron Bernstein, Chengyuan Deng, Jie Gao, and Nicole Wein. Low sensitivity hopsets. *arXiv preprint*, 2024. doi:10.48550/arXiv.2407.10249.
5. Felix A Behrend. On sets of integers which contain no three terms in arithmetical progression. *Proceedings of the National Academy of Sciences*, 32(12):331–332, 1946.
6. Uri Ben-Levy and Merav Parter. New (α, β) spanners and hopsets. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1695–1714. SIAM, 2020. doi:10.1137/1.9781611975994.104.
7. Aaron Bernstein, Maximilian Probst Gutenberg, and Christian Wulff-Nilsen. Near-optimal decremental SSSP in dense weighted digraphs. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1112–1122. IEEE, 2020. doi:10.1109/FOCS46700.2020.00107.
8. Aaron Bernstein and Nicole Wein. Closing the gap between directed hopsets and shortcut sets. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 163–182. SIAM, 2023. doi:10.1137/1.9781611977554.CH7.

9 Greg Bodwin. New results on linear size distance preservers. *SIAM Journal on Computing*, 50(2):662–673, 2021. doi:10.1137/19M123662X.

10 Greg Bodwin, Chengyuan Deng, Jie Gao, Gary Hoppenworth, Jalaj Upadhyay, and Chen Wang. The discrepancy of shortest paths. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*, volume 297 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:20, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPICS.ICALP.2024.27.

11 Greg Bodwin and Gary Hoppenworth. Folklore sampling is optimal for exact hopsets: Confirming the $\sqrt{n}$ barrier. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 701–720. IEEE, 2023.

12 Nairen Cao, Jeremy T Fineman, and Katina Russell. Efficient construction of directed hopsets and parallel approximate shortest paths. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 336–349, 2020. doi:10.1145/3357713.3384270.

13 Nairen Cao, Jeremy T Fineman, and Katina Russell. Brief announcement: An improved distributed approximate single source shortest paths algorithm. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, pages 493–496, 2021. doi:10.1145/3465084.3467945.

14 Shiri Chechik and Tianyi Zhang. Path-reporting distance oracles with logarithmic stretch and linear size. In *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*, pages 42:1–42:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.ICALP.2024.42.

15 Justin Y. Chen, Badih Ghazi, Ravi Kumar, Pasin Manurangsi, Shyam Narayanan, Jelani Nelson, and Yinzhan Xu. Differentially private all-pairs shortest path distances: Improved algorithms and lower bounds. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 5040–5067, 2023. doi:10.1137/1.9781611977554.CH184.

16 Edith Cohen. Polylog-time and near-linear work approximation scheme for undirected shortest paths. *Journal of the ACM (JACM)*, 47(1):132–166, 2000. doi:10.1145/331605.331610.

17 Don Coppersmith and Michael Elkin. Sparse sourcewise and pairwise distance preservers. *SIAM Journal on Discrete Mathematics*, 20(2):463–501, 2006. doi:10.1137/050630696.

18 Chengyuan Deng, Jie Gao, Jalaj Upadhyay, and Chen Wang. Differentially private range query on shortest paths. In *Algorithms and Data Structures Symposium*, pages 340–370. Springer, 2023.

19 Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014. doi:10.1561/0400000042.

20 Michael Elkin and Ofer Neiman. Linear-size hopsets with small hopbound, and distributed routing with low memory. *arXiv preprint*, 2017. arXiv:1704.08468.

21 Michael Elkin and Ofer Neiman. Hopsets with constant hopbound, and applications to approximate shortest paths. *SIAM Journal on Computing*, 48(4):1436–1480, 2019. doi:10.1137/18M1166791.

22 Michael Elkin and Idan Shabat. Path-reporting distance oracles with logarithmic stretch and size $o(n \log \log n)$. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2278–2311. IEEE, 2023. doi:10.1109/FOCS57990.2023.00141.

23 Chenglin Fan and Ping Li. Distances release with differential privacy in tree and grid graph. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pages 2190–2195. IEEE, 2022. doi:10.1109/ISIT50566.2022.9834836.

24 Jeremy T Fineman. Nearly work-efficient parallel algorithm for digraph reachability. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 457–470, 2018. doi:10.1145/3188745.3188926.

25 Sebastian Forster and Danupon Nanongkai. A faster distributed single-source shortest paths algorithm. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 686–697. IEEE, 2018. doi:10.1109/FOCS.2018.00071.

26 Maximilian Probst Gutenberg and Christian Wulff-Nilsen. Decremental SSSP in weighted digraphs: Faster and against an adaptive adversary. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2542–2561. SIAM, 2020. doi:10.1137/1.9781611975994.155.

27 Dov Harel and Robert Endre Tarjan. Fast algorithms for finding nearest common ancestors. *siam Journal on Computing*, 13(2):338–355, 1984. doi:10.1137/0213024.

28 Monika Henzinger, Sebastian Krinninger, and Danupon Nanongkai. Sublinear-time decremental algorithms for single-source reachability and shortest paths on directed graphs. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 674–683, 2014. doi:10.1145/2591796.2591869.

29 Monika Henzinger, Sebastian Krinninger, and Danupon Nanongkai. Improved algorithms for decremental single-source reachability on directed graphs. In *Automata, Languages, and Programming: 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I 42*, pages 725–736. Springer, 2015. doi:10.1007/978-3-662-47672-7_59.

30 William Hesse. Directed graphs requiring large numbers of shortcuts. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '03, pages 665–669, USA, 2003. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=644108.644216.

31 Shang-En Huang and Seth Pettie. Thorup–Zwick emulators are universally optimal hopsets. *Information Processing Letters*, 142:9–13, 2019. doi:10.1016/J.IPL.2018.10.001.

32 Shang-En Huang and Seth Pettie. Lower bounds on sparse spanners, emulators, and diameter-reducing shortcuts. *SIAM Journal on Discrete Mathematics*, 35(3):2129–2144, 2021. doi:10.1137/19M1306154.

33 Arun Jambulapati, Yang P Liu, and Aaron Sidford. Parallel reachability in almost linear work and square root depth. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1664–1686. IEEE, 2019. doi:10.1109/FOCS.2019.00098.

34 Adam Karczmarz and Piotr Sankowski. A deterministic parallel APSP algorithm and its applications. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 255–272. SIAM, 2021. doi:10.1137/1.9781611976465.17.

35 Philip N Klein and Sairam Subramanian. A randomized parallel algorithm for single-source shortest paths. *Journal of Algorithms*, 25(2):205–220, 1997. doi:10.1006/JAGM.1997.0888.

36 Shimon Kogan and Merav Parter. Beating matrix multiplication for $n^{1/3}$-directed shortcuts. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, pages 82:1–82:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ICALP.2022.82.

37 Shimon Kogan and Merav Parter. Having hope in hops: New spanners, preservers and lower bounds for hopsets. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 766–777. IEEE, 2022. doi:10.1109/FOCS54457.2022.00078.

38 Shimon Kogan and Merav Parter. New diameter-reducing shortcuts and directed hopsets: Breaking the barrier. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1326–1341. SIAM, 2022. doi:10.1137/1.9781611977073.55.

39 Kevin Lu, Virginia Vassilevska Williams, Nicole Wein, and Zixuan Xu. Better lower bounds for shortcut sets and additive spanners via an improved alternation product. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3311–3331. SIAM, 2022. doi:10.1137/1.9781611977073.131.

40 Idan Shabat and Ofer Neiman. A unified framework for hopsets and spanners. *arXiv preprint arXiv:2108.09673*, 2021. arXiv:2108.09673.

41 Hanmao Shi and Thomas H Spencer. Time–work tradeoffs of the single-source shortest paths problem. *Journal of algorithms*, 30(1):19–32, 1999. doi:10.1006/JAGM.1998.0968.

42 Endre Szemerédi and William T. Trotter. Extremal problems in discrete geometry. *Combinatorica*, 3(3):381–392, 1983. doi:10.1007/BF02579194.

43 Sasu Tarkoma. *Overlay Networks: Toward Information Networking*. CRC Press, February 2010.

44 Mikkel Thorup. On shortcutting digraphs. In *Graph-Theoretic Concepts in Computer Science: 18th International Workshop, WG'92 Wiesbaden-Naurod, Germany, June 18–20, 1992 Proceedings 18*, pages 205–211. Springer, 1993.

45 Mikkel Thorup and Uri Zwick. Spanners and emulators with sublinear distance errors. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete Algorithm*, pages 802–809, 2006. URL: http://dl.acm.org/citation.cfm?id=1109557.1109645.

46 Jeffery Ullman and Mihalis Yannakakis. High-probability parallel transitive closure algorithms. In *Proceedings of the second annual ACM symposium on Parallel Algorithms and Architectures*, pages 200–209, 1990.

47 Virginia Vassilevska Williams, Yinzhan Xu, and Zixuan Xu. Simpler and higher lower bounds for shortcut sets. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2643–2656. SIAM, 2024. doi:10.1137/1.9781611977912.94.

The More the Merrier! On Total Coding and Lattice Problems and the Complexity of Finding Multicollisions*

Huck Bennett
University of Colorado Boulder, CO, USA

Surendra Ghentiyala
Cornell University, Ithaca, NY, USA

Noah Stephens-Davidowitz
Cornell University, Ithaca, NY, USA

Abstract

We show a number of connections between two types of search problems: (1) the problem of finding an L-wise *multicollision* in the output of a function; and (2) the problem of finding two codewords in a code (or two vectors in a lattice) that are within distance d of each other. Specifically, we study these problems in the *total* regime, in which L and d are chosen so that such a solution is guaranteed to exist, though it might be hard to find.

In more detail, we study the total search problem in which the input is a function $\mathcal{C} : [A] \to [B]$ (represented as a circuit) and the goal is to find $L \leq \lceil A/B \rceil$ *distinct* elements $x_1, \ldots, x_L \in A$ such that $\mathcal{C}(x_1) = \cdots = \mathcal{C}(x_L)$. The associated complexity classes Polynomial Multi-Pigeonhole Principle ((A, B)-PMPP^L) consist of all problems that reduce to this problem.

We show close connections between (A, B)-PMPP^L and many celebrated upper bounds on the minimum distance of a code or lattice (and on the list-decoding radius). In particular, we show that the associated computational problems (i.e., the problem of finding two distinct codewords or lattice points that are close to each other) are in (A, B)-PMPP^L, with a more-or-less smooth tradeoff between the distance d and the parameters A, B, and L. These connections are particularly rich in the case of codes, in which case we show that multiple incomparable bounds on the minimum distance lie in seemingly incomparable complexity classes.

Surprisingly, we also show that the computational problems associated with some bounds on the minimum distance of codes are actually *hard* for these classes (for codes represented by arbitrary circuits). In fact, we show that finding two vectors within a certain distance d is actually hard for the important (and well-studied) class $\mathsf{PWPP} = (B^2, B)$-PMPP^2 in essentially all parameter regimes for which an efficient algorithm is not known, so that our hardness results are essentially tight. In fact, for some d (depending on the block length, message length, and alphabet size), we obtain both hardness and containment. We therefore completely settle the complexity of this problem for such parameters and add coding problems to the short list of problems known to be complete for PWPP.

We also study (A, B)-PMPP^L as an interesting family of complexity classes in its own right, and we uncover a rich structure. Specifically, we use recent techniques from the cryptographic literature on multicollision-resistant hash functions to (1) show inclusions of the form (A, B)-$\mathsf{PMPP}^L \subseteq (A', B')$-$\mathsf{PMPP}^{L'}$ for certain non-trivial parameters; (2) black-box separations between such classes in different parameter regimes; and (3) a non-black-box proof that (A, B)-$\mathsf{PMPP}^L \in \mathsf{FP}$ if (A', B')-$\mathsf{PMPP}^{L'} \in \mathsf{FP}$ for yet another parameter regime. We also show that (A, B)-PMPP^L lies in the recently introduced complexity class Polynomial Long Choice for some parameters.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

* An earlier version of this work used the title "The more the merrier! On the complexity of finding multicollisions, with connections to codes and lattices."



© Huck Bennett, Surendra Ghentiyala, and Noah Stephens-Davidowitz;
licensed under Creative Commons License CC-BY 4.0

16th Innovations in Theoretical Computer Science Conference (ITCS 2025).
Editor: Raghu Meka; Article No. 14; pp. 14:1–14:22



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Keywords and phrases Multicollisions, Error-correcting codes, Lattices

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.14

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2024/018/ [4]

Funding *Huck Bennett*: This work is supported in part by NSF Grant No. 2312297. Part of this work was done while the author was at Oregon State University.
Surendra Ghentiyala: This work is supported in part by the NSF under Grants Nos. CCF-2122230 and CCF-2312296, a Packard Foundation Fellowship, and a generous gift from Google.
Noah Stephens-Davidowitz: This work is supported in part by the NSF under Grants Nos. CCF-2122230 and CCF-2312296, a Packard Foundation Fellowship, and a generous gift from Google. Some of this work was completed while the author was visiting the National University of Singapore and the Centre for Quantum Technologies.

Acknowledgements The authors would like to thank Atri Rudra for very helpful discussions.

1 Introduction

We study *the complexity of total coding problems*, as well as total lattice problems. (We focus on coding problems in the introduction.)

Recall that a q-ary code with messages of length k, block length n, and distance d is a function $\mathcal{C} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ such that

$$d = \min_{\boldsymbol{x}_1 \neq \boldsymbol{x}_2} \Delta(\mathcal{C}(\boldsymbol{x}_1), \mathcal{C}(\boldsymbol{x}_2)) \ ,$$

where Δ is the Hamming distance, i.e., the number of entries in which two elements in $\mathbb{F}_q^n$ differ. For our purposes, we think of $\mathcal{C}$ as an arbitrary circuit with size $\mathrm{poly}(n, k, \log q)$. (Much of the literature is concerned with the special case when $\mathcal{C}$ is a non-singular *linear* function, in which case the code is called a *linear* code. We discuss our choice of definition more in Section 1.5.)

The most fundamental question in coding theory is to find bounds on d for fixed n, k, and q. In other words, one wishes to argue that any set of q^k points in $\mathbb{F}_q^n$ must contain two points that are within Hamming distance d for the smallest value of d possible. Many beautiful upper bounds on the distance d are known in terms of n, k, and q.

We therefore define the natural family of computational search problem associated with this question. Specifically, we define the $(n, k, d)_q$-Short Distance Problem ($(n, k, d)_q$-SDP) as the computational search problem in which the input is a circuit $\mathcal{C} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ and the goal is to find $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$ such that $\Delta(\mathcal{C}(\boldsymbol{x}_1), \mathcal{C}(\boldsymbol{x}_2)) \leq d$.[1] Notice that $(n, k, d)_q$-SDP is total if and only if all q-ary codes with message length k and block length n have distance at most d.

So, upper bounds on the minimum distance of a code correspond to proofs of *totality* of SDP for some choice of $d = d(n, k, q)$. In other words, such bounds can be viewed as proofs that for such a choice of d, SDP is in the complexity class TFNP, which consists of *total* search problems with efficiently verifiable solutions. Starting with the seminal work of

[1] The problem of finding $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$ that minimizes $\Delta(\mathcal{C}(\boldsymbol{x}_1), \mathcal{C}(\boldsymbol{x}_2))$ for the given input $\mathcal{C}$ is called the Minimum Distance Problem (MDP), and it is known to be NP-hard, even to approximate, and even for linear codes [14]. In contrast, SDP only asks for two codewords within some fixed distance, independent of the minimum distance of the code. Furthermore, one is often interested only in the important special case in which $\mathcal{C}$ computes a linear function over $\mathbb{F}_q$, but we stick to this more general setting because our upper bounds for this problem apply even to arbitrary codes (making them stronger), and unfortunately our lower bounds seem to require us to work with arbitrary codes specified by circuits.

Papadimitriou [34], the study of TFNP has focused largely on placing important total search problems in subclasses of TFNP, where it is often convenient to think of each subclass as capturing problems whose totality can be attributed to a particular fundamental principle. E.g., PPP (which we will discuss much more shortly) can be thought of as the subclass of TFNP corresponding to problems whose totality can be proven via a particular version of the pigeonhole principle.

It is therefore natural to ask what more we can say about the complexity of $(n, k, d)_q$-SDP when $d = d(n, k, q)$ is equal to one of the many celebrated upper bounds on the minimum distance of a q-ary code with block length n and message length k. Indeed, there are now many important upper bounds known on the minimum distance of codes (in terms of n, k, and q), such as the Singleton bound [39], the Hamming bound [19], the Plotkin bound [36], the Elias-Bassalygo bound [3], and the MRRW bound(s) [29]. These bounds are ubiquitous in the computer science literature and beyond. (See, e.g., [18].)

However, in spite of the importance of these bounds, there has been surprisingly little work addressing the complexity of the associated (total) search problems of actually finding a pair of distinct codewords within one of these bounds. Indeed, to our knowledge, the only prior work directly addressing such a question is the beautiful (and recent) work of Debris-Alazard, Ducas, and van Woerden [11], which showed (among other things) that the problem of solving SDP on linear codes within the Griesmer bound [16] can be solved efficiently.

1.1 The problem of finding multicollisions

In this work, we will show connections between the complexity of such total coding (and lattice) problems and a natural family of total search problems.

Specifically, for integers $A > B$ and $L \geq 2$, we consider the computational problem in which the input is a circuit $\mathcal{C} : [A] \to [B]$,[2] and the goal is to find L distinct input values $x_1, x_2, \ldots, x_L$ such that $\mathcal{C}(x_1) = \mathcal{C}(x_2) = \cdots = \mathcal{C}(x_L)$. Notice that, by the (generalized) pigeonhole principle, this problem is total if (and only if) the size A of the domain of $\mathcal{C}$ and the size B of its range satisfy $L \leq \lceil A/B \rceil$. We will focus on the regime in which the problem is total.

In the special case when $L = 2$ and $A = B + 1$, this is the canonical complete problem for the Polynomial Pigeonhole Principle complexity class (PPP).[3] This class was introduced by Papadimitriou in his work studying problems in TFNP (i.e., total search problems in FNP) [34]. Since then, the class PPP has been of great interest because it is known to contain many important computational problems, such as the problem of breaking a cryptographic collision-resistant hash function, the problem of breaking a one-way permutation, factoring (under randomized reductions) [22], and many more problems of interest [41, 8]. Most relevantly to our work, the problem of finding a non-zero vector in a lattice within Minkowski's bound (in the ℓ_∞ norm, though see below) was shown to be in PPP in [2].

Because of its relationship with the pigeonhole principle, we call the computational problem of finding such a collision Pigeon. (Papadimitriou originally used this name for the special case when $L = 2$ [34].) In fact, we get a family of problems (A, B)-Pigeon^L,

[2] See the full version [4] for discussion of what we mean by a circuit with input size A and output size B when A and B are not necessarily powers of 2.

[3] When $L = 2$ and $A/B \geq 1 + 1/\operatorname{poly}(\log B)$, one obtains the canonical complete problem for Polynomial Weak Pigeonhole Principle complexity class (PWPP). We will say much more about this distinction later, but for now we largely ignore this.

parameterized by the circuit input size A, the circuit output size B, and the number of colliding inputs $L \leq \lceil A/B \rceil$ that we must find. Accordingly, we define the complexity class (A, B)-PMPP^L as the set of all search problems that have a polynomial-time (Karp) reduction to (A, B)-Pigeon^L.[4]

In the complexity-theoretic literature, there is very little work on Pigeon for $L > 2$ (although we note Sotiraki's thesis [40, Section 4.5] as foundational work in this direction; and see also Section 1.3 for discussion of recent independent work by [21]). On the other hand, there is an exciting line of work in the cryptographic literature studying *multicollision-resistant hash functions* [23, 33, 43, 5, 28, 6, 25, 40, 13, 37, 20]. From our perspective, one can think of such cryptographic works as studying the *average-case* complexity of the Pigeon problem (under efficiently sampleable distributions of circuits $\mathcal{C}$). However, even these works have not fully explored this space, primarily focusing on the case where L is constant and $\log A \gtrsim 2 \log B$.

1.2 Our results

In this work, we show many connections between coding and lattice problems and PMPP. We also further study the classes (A, B)-PMPP^L.

1.2.1 Connections between coding problems and PMPP

Our first set of results shows a number of connections between PMPP and computational search problems related to error-correcting codes. (See Figure 1 for a summary of some of our results for binary codes.)

1.2.1.1 Upper bounds on the Shortest Distance Problem

We show that many of the celebrated bounds on the minimum distance of a code yield versions of SDP that are in different versions of PMPP, including the Singleton bound [39], the Hamming bound [19], the Plotkin bound [36], and the Elias-Bassalygo bound [3]. The following theorem shows some of these results.

▶ **Theorem 1** (Informal, see full version [4]). *The following hold for any n, k, and prime power q and any constant $\varepsilon > 0$.*

1. (n, k, d_S)-SDP$_q$ *is in* PWPP *for* $d_S(n, k, q) := n - k + 1$ *equal to the Singleton bound.*
2. (n, k, d_H)-SDP$_q$ *is in* PPP *and* $(n, k, (1 + \varepsilon)d_H)$-SDP$_q$ *is in* PWPP *for* $d_H = d_H(n, k, q)$ *equal to the Hamming bound.*
3. (n, k, d_P)-SDP$_q$ *is in* PMPP *for* $d_P(n, k, q) = (1 - 1/q + \varepsilon)(n - k)$ *slightly larger than the Plotkin bound.*
4. (n, k, d_E)-SDP$_q$ *is in* PMPP *for* $d_E(n, k, q)$ *equal to the Elias-Bassalygo bound.*

In fact, these results are special cases of more general results that we prove. Specifically, we show that one can obtain smaller values of d by increasing L *or* decreasing A (while holding B fixed and maintaining totality).

Our smooth tradeoff in particular is perhaps a bit surprising. Indeed, it is a-priori not clear why it would be useful to take $L > 2$ in such a reduction, since, after all, the goal in SDP is to find *two* codewords. (Of course, this is because our reductions mimic the proofs of

[4] To define PMPP formally, A and B should be functions of some asymptotic parameter n, and L might also be a function of n. But, we mostly ignore this issue in the introduction for simplicity.

the Plotkin and Elias-Bassalygo bounds. Both proofs work by first finding $L > 2$ codewords that lie in a relatively small Hamming ball and then showing that two of these codewords must be quite close to each other.)

1.2.1.2 Tight hardness for SDP and completeness when $L = 2$

We next show tight hardness results for SDP. In particular, we consider both the case when a code is represented by an arbitrary circuit $\mathcal{C}$ *and* the case when a code is represented in systematic form (i.e., the case when the first k symbols in a codeword consist of the message; see Section 1.5). We call this restricted problem sysSDP.

▶ **Theorem 2** (Informal, see full version [4]). *For any n, k, and prime power q and any constant $\varepsilon > 0$, (n, k, d)-SDP$_q$ is* PWPP*-hard for $d(n, k, q) = (1 - 1/q - \varepsilon)n$. Furthermore, (n, k, d_P)-sysSDP$_q$ is* PWPP*-hard for distance $d_P(n, k, q) = (1-1/q-\varepsilon)(n-k)$ slightly below the Plotkin bound.*

Theorem 2 is essentially tight. In particular, for $d \geq (1 - 1/q)n$, it is easy to see that SDP can be solved efficiently. And, we show that there is also a simple efficient algorithm for sysSDP for $d \gtrsim (1-1/q)(n-k)$ (see full version [4]). So, Theorem 2 essentially characterizes when there is a polynomial-time algorithm for SDP and sysSDP (assuming that FP $\neq$ PWPP).

Furthermore, there is a large overlap between the parameter regime for which we show containment in PWPP and the regime for which we show PWPP-hardness (both for arbitrary codes and codes in systematic form), so that for a large range of parameters we show that SDP is actually *complete* for PWPP. This essentially settles the complexity of SDP in these regimes and adds SDP to the relatively short list of problems known to be complete for this class. For example, we show that in many settings, the Hamming bound is PWPP-complete. (Since the Hamming bound can be viewed as a coding-theoretic analogue of Minkowski's bound for lattices, one might view this result as resolving a coding-theoretic analogue of the conjecture that Minkowski's bound is PPP-complete [2]. However, perhaps a better coding-theoretic analogue would be PPP-hardness of this problem over *linear* codes, which we do not prove. See Section 1.6.)

We summarize these results for binary codes in Figure 1. The picture for q-ary codes is quite similar (with the various bounds replaced by their q-ary versions). See full version [4] for the details.

1.2.1.3 Finding *many* close codewords and PMPP

We also show analogous results for the problem of finding *many* codewords that all lie in a small ball. This problem corresponds to bounds on the *list decoding* radius of a code (just as the problem of finding close pairs of codewords corresponds to bounds on the *unique decoding* radius).

We first show that list-decoding generalizations of the Singleton bound and the Hamming bound lie in PMPP for appropriate parameters. (See full version [4]) We then show that the problem of finding many codewords that all lie in a small ball is also *hard* for PMPP for some choices of parameters.

Our results for this problem are significantly more subtle than the $L = 2$ case, since for $L > 2$, the complexity of (A, B)-Pigeon^L seems to vary quite a bit with the parameters, whereas for $L = 2$, the parameters A and B matter much less. In particular, for $L > 2$ we do not find a complete problem for PMPP^L. See full version [4] for the details.

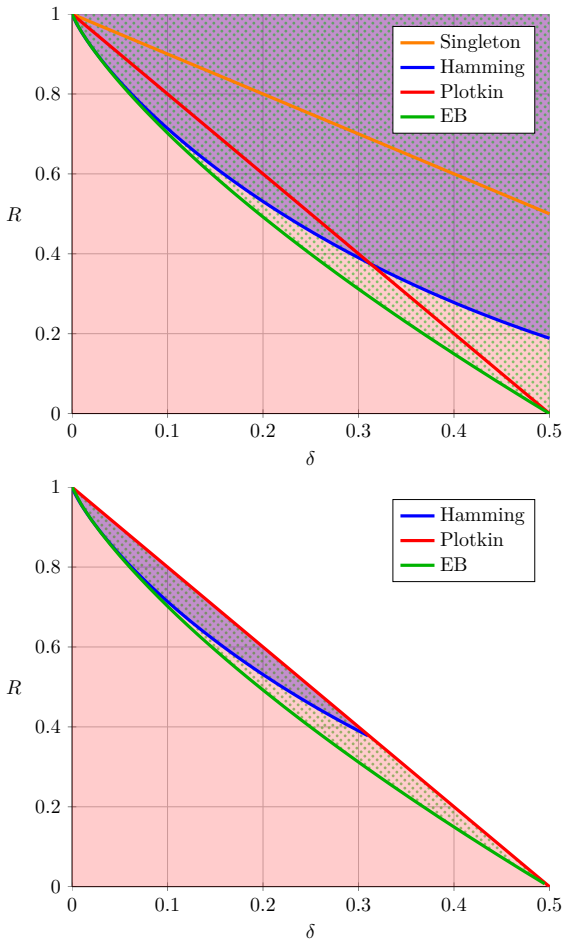


Figure 1 Two plots showing the complexity of the problem of finding two codewords within relative distance $\delta := d/n$ in a code with rate $R := k/n$ for binary codes (represented by circuits). The shaded red region represents PWPP-hardness. (The entirety of both figures are shaded red.) Regions not shown ($\delta > 1/2$ for the left figure and $\delta > 1 - 2R$ for the right figure) are known to be in FP. The shaded blue region represents containment in PWPP. The region of overlap between red and blue therefore represents regimes where the problem is PWPP-complete. The area covered with green dots represents problems in PMPP^L for some $L \geq 2$. The left figure is for arbitrary (binary) codes represented by arbitrary circuits, while the right figure is for (binary) codes in systematic form. (See Section 1.5 for discussion of systematic form and how we define codes in this context.)

1.2.2 Lattice problems in PMPP

We next show similar results for computational lattice problems. Recall that a lattice is the set of all integer linear combinations of n linearly independent basis vectors $\mathbf{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \in \mathbb{Z}^n$, i.e.,

$$\mathcal{L} = \mathcal{L}(\mathbf{B}) = \{z_1\boldsymbol{b}_1 + \cdots + z_n\boldsymbol{b}_n \ : \ z_i \in \mathbb{Z}\} \ .$$

A fundamental question about lattices asks when there must exist a non-zero lattice point $\boldsymbol{y} \in \mathcal{L}_{\neq \mathbf{0}}$ such that $\|\boldsymbol{y}\|_K \leq r$, where $\|\cdot\|_K$ is some norm of interest and r is some bound.

Minkowski's celebrated theorem [32] tells us that such a $\boldsymbol{y}$ is guaranteed to exist for some

$$r \leq 2\det(\mathcal{L})^{1/n}/\operatorname{vol}(K)^{1/n} ,$$

where $\det(\mathcal{L}) := |\det(\mathbf{B})|$ is the lattice determinant and K is the unit ball of the norm $\|\cdot\|_K$. The corresponding computational problem Minkowski_K is the search problem in which the input is a basis $\mathbf{B}$ for a lattice, and the goal is to output a non-zero vector $\boldsymbol{y} \in \mathcal{L}_{\neq \mathbf{0}}$ such that

$$\|\boldsymbol{y}\|_K \leq 2\det(\mathcal{L})^{1/n}/\operatorname{vol}(K)^{1/n} .$$

This problem is quite important in cryptography, particularly in the ℓ_2 norm. In the special case of the ℓ_∞ norm, it is known that $\text{Minkowski}_\infty \in \mathsf{PPP}$, and Ban, Jain, Papadimitriou, Psomas, and Rubinstein [2] conjectured that Minkowski_∞ is actually PPP-complete. (This conjecture remains open.)

We study the more general problem of finding $\boldsymbol{y} \in \mathcal{L}_{\neq \mathbf{0}}$ with

$$\|\boldsymbol{y}\|_p \leq \gamma \det(\mathcal{L})^{1/n} ,$$

where

$$\|\boldsymbol{y}\|_p := (|y_1|^p + \cdots + |y_n|^p)^{1/p}$$

is the ℓ_p norm. This problem is known as the γ-Hermite Shortest Vector Problem (γ-HSVP_p). Since this problem is relevant to cryptography, it is very well studied for a wide range of parameters γ [27, 38, 15, 31, 1], particularly for $p = 2$.

The case $\gamma = 2/\operatorname{vol}(\mathcal{B}_p^n)^{1/n}$ corresponds to Minkowski_p, where $\mathcal{B}_p^n$ is the unit ℓ_p ball. However, Minkowski's bound is not tight (except in the case when K tiles space, such as when K is the hypercube). For example, Blichfeldt improved on Minkowski's celebrated theorem in the ℓ_2 norm, proving that γ-HSVP_2 is still total when $\gamma \approx \sqrt{2}/\operatorname{vol}(\mathcal{B}_2^n)^{1/n}$ [7].[5]

Perhaps surprisingly, we show that γ-$\text{HSVP}_2 \in (A, B)$-PMPP^L for $\gamma \approx \sqrt{2}/\operatorname{vol}(\mathcal{B}_2^n)^{1/n}$ and appropriate choices of A, B, and L. In fact, we show a smooth tradeoff between the Hermite factor γ and the parameters A, B, and L, showing that one can obtain shorter vectors by either increasing L or decreasing the ratio between A and B (while maintaining totality). A similar story holds for ℓ_p norms more generally.

▶ **Theorem 3** (Informal; see full version [4]). *For any constant integer $p \geq 1$ and any $A, B, L \in \mathbb{Z}^+$ satisfying $2^{s(n)} = B < A \leq 2^{\text{poly}(n)}$ and $L \leq \lceil A/B \rceil$ for a sufficiently large polynomial $s(n)$, it holds that γ-$\text{HSVP}_p \in (A, B)$-PMPP^L for*

$$\gamma \approx d_p(L, n) \cdot (A/B)^{1/n} \cdot \operatorname{vol}(\mathcal{B}_p^n)^{-1/n} ,$$

where n is the rank of the lattice in the γ-HSVP instance and $1 \leq d_p(L, n) \leq 2$ is a particular function that is decreasing in the parameter L.

[5] We note that Blichfeldt proved two distinct relevant theorems in this context, which might easily confuse the reader. So, we endeavor to clarify here. The theorem commonly referred to as "Blichfeldt's theorem" says that any measurable set $S \subset \mathbb{R}^n$ with $\operatorname{vol}(S) > \det(\mathcal{L})$ is guaranteed to contain two points $\boldsymbol{x}_1, \boldsymbol{x}_2 \in S$ with $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$ such that $\boldsymbol{x}_1 - \boldsymbol{x}_2 \in \mathcal{L}$. This theorem is often discussed in the context of total lattice problems because it can be used to prove Minkowski's theorem. In fact, Sotiraki, Zampetakis, and Zirdelis introduced a related computational problem that they called BLICHFELDT (in which the set S is represented implicitly by a circuit), and they showed that BLICHFELDT is actually PPP-complete. It is not clear how BLICHFELDT is related to γ-HSVP for $\gamma \approx \sqrt{2}/\operatorname{vol}(\mathcal{B}_2^n)^{1/n}$ (except that they are two computational lattice problems whose totality was proven by Blichfeldt).

1.2.3 Containments between different classes

The above results bring new attention to the classes PMPP. So, we next study containments between these PMPP classes with different parameters A, B, and L, and other classes of interest in TFNP.

Recall that the celebrated Merkle-Damgård construction [30, 10] shows that the ratio of the input size A to the output size B of a circuit essentially does not matter in the special case when $L = 2$, since one can efficiently reduce from the problem of finding a single collision (i.e., $L = 2$) in a barely compressing circuit $\mathcal{C} : A \to B$ with $A = B + B/\operatorname{poly}(\log B)$ to the (seemingly much easier) problem of finding a single collision in a much more compressing circuit $\mathcal{C}' : A' \to B$ with $\log A' = \log(B)^C$ for any constant $C > 1$. In our terminology,

$$(B + B/\operatorname{poly}(\log B), B)\text{-PMPP}^2 = (2^{\log^C B}, B)\text{-PMPP}^2 .$$

This surprising collapse of complexity classes is known as *domain extension*, and it has innumerable applications in cryptography and complexity theory.[6]

Already in 2004, Joux noticed that Merkle and Damgård's elegant domain extension technique does not seem to work for $L > 2$ [23]. So, it appears that for $L > 2$, the relationship between A and B (i.e., how "compressing" the circuit $\mathcal{C}$ is) might matter quite a bit. This suggests a surprising fundamental difference between the case $L = 2$ and the case when $L > 2$.

However, we show two (rather weak) notions of domain extension that still work in the setting of multicollisions. The first such result follows by analyzing the Merkle-Damgård construction in this setting and showing that it does achieve *something*, albeit with a large loss in parameters. The second result shows a more sophisticated reduction (using Merkle trees together with list-recoverable codes) that is better than the Merkle-Damgård-based reduction in the regime where A is very large compared to B. The latter result can be thought of as a translation into our setting of a beautiful result due to Bitanski, Kalai, and Paneth [6] in the setting of multicollision-resistant hash functions. (In fact, the proof is substantially simpler in our setting than in that of [6], since we do not have to worry about the many additional issues that arise in the average-case setting.)

Together, these results show that it is possible to reduce the problem of finding multicollisions in a less compressing function to the problem of finding multicollisions in a more compressing function, but at the expense of a large loss in the number of collisions found.

▶ **Theorem 4** (Informal; see full version [4]). *For $m > a > b$,*

$$(2^a, 2^b)\text{-PMPP}^{L'} \subseteq (2^m, 2^b)\text{-PMPP}^L$$

for $L' \approx L^{(a-b)/(m-b)}$.

For any $r \geq 2$, any $k \geq 1$, and any L,

$$(2^{vr}, 2^v)\text{-PMPP}^{M'} \subseteq (2^{vrk}, 2^v)\text{-PMPP}^M$$

under randomized reductions, for

$$M' \approx M^{\log r/(2\log r + \log k + \log(M)/2)} .$$

[6] Admittedly, we are deliberately conflating the distinction between the problem of breaking a cryptographic hash function (which is what Merkle and Damgård actually studied) and the problem of finding a collision in an arbitrary worst-case circuit. All of the above statements hold in both cases.

We also show that the class PMPP is contained in the recently introduced ([35]) complexity class Polynomial Long Choice (PLC), for appropriate choices of the parameters A, B, and L. In fact, we show a reduction to the Unary Long Choice problem. (This result was recently independently discovered in [21]. See Section 1.3.) This strengthens a result of Pasarkar, Papadimitriou, and Yannakakis [35], who showed that PWPP $\subseteq$ PLC.

▶ **Theorem 5** (Informal; see full version [4]). *For any $L < n$,*

$(2^n, 2^{n-L})$-PMPP$^L \subseteq$ PLC

1.2.4 Black-box separations (and a non-black-box non-separation)

Our final set of results concerns black-box separations between (A, B)-PMPP^L for different values of A, B, and L, which suggest that it might be hard to prove stronger containments than what we show above. However, we note that recent independent work of Jain, Li, Robere, and Xun [21] also showed exciting black-box separations of this form. While our results are formally incomparable to theirs, we believe that the results of [21] are more interesting than our own. (See Section 1.3.)

The starting point for our results is a beautiful idea due to Komargodski, Naor, and Yogev [25] for separating $(2^n, 2^{n/2})$-PMPP^L from $(2^n, 2^{n/2})$-PMPP$^{L'}$ for any constants $L \neq L'$ (particularly in the average-case setting relevant to cryptography). Unfortunately, however, the proof in [25] had a subtle bug that does not seem easy to fix [26].

We use similar ideas to show two different black-box separations, which can be seen as partial evidence that domain extension and range compression are not possible when $L > 2$. However, we note that our black-box separations are rather weak, since they only rule out rather fine-grained black-box reductions between the classes.

Finally, we show that a very clever non-black-box proof due to Rothblum and Vasudevan [37] extends to our setting. In particular, Rothblum and Vasudevan show, using non-black-box techniques, that the existence of a certain sufficiently strong multicollision-resistant hash function implies the existence of a collision-resistant hash function. We prove an analogue of their result in our (worst-case) setting, showing that suitable hardness of PMPP for large L implies hardness of PMPP for smaller L. (After a preliminary version of this work appeared, Buzek and Tessaro [9] improved upon the main result in [37]. We do not know whether the stronger result in [9] extends to our setting.)

As these results are rather technical, we refer the reader to the full version [4] for the details.

1.3 Comparison with Jain, Li, Robere, and Xun

Recent exciting work by Jain, Li, Robere, and Xun [21] also defines and studies the computational problem (A, B)-Pigeon^L and the associated complexity class (A, B)-PMPP^L (with slightly different notation). (They also define additional classes that correspond to the union of (A, B)-PMPP^L over different parameters A, B, and L.) [21] is concurrent with and independent of this work (and appeared as a preprint shortly before we released a preprint of the present work). Here, we provide a brief comparison of their work with ours.

Both the present work and [21] define multi-collision classes and study relationships between them. At a high level, [21] has morally stronger (although formally incomparable) results on the structural complexity of these classes, whereas the present work focuses on showing containment and hardness of coding and lattice problems with respect to these classes (which [21] does not study at all).

In terms of structural complexity, [21] contains exciting black-box separation results, which, although formally incomparable to our black-box separations, we think of as stronger. In particular, [21] are essentially able to black-box separate (A, B)-PMPP^L from (A', B')-PMPP$^{L'}$ for *any* constants $L \neq L'$ and *any* (reasonable) A, B, A', and B'. They also show black-box separations between PMPP and other interesting complexity classes in TFNP, and in particular show a black-box separation between the Ramsey problem and PMPP. We refer the reader to [21] for the technical details.

[21] also studies the relationship between PLC and PMPP. Indeed, they prove a result that is essentially identical to our Theorem 5, which shows that PMPP $\subseteq$ PLC for certain parameters. (Formally, our technical result in the full version [4] is more general than the analogous result in [21], but it is clear that the proof in [21] yields the more general result as well.) In addition, they show a containment in the other direction, that PLC $\subseteq$ PMPP, albeit for different parameters. Finally, [21] shows that an interesting problem in TFBQP is in PMPP. The latter two results have no analogue in the present work.

In this work, we focus on the relationship of PMPP with coding and lattice problems (see the full version [4]). We also show Merkle-Damgård-style reductions between PMPP with different parameters (see full version [4]) and a non-block-box non-separation in the style of [37] (see full version [4]). These topics are not studied in [21].

1.4 Other related work

Our work lies at the intersection of a number of different areas, and there is therefore much related work to discuss in addition to [21]. Here, we focus on how this prior work relates to our work.

The complexity of total lattice problems

The complexity of Minkowski's bound and HSVP more generally is quite well studied, particularly in the ℓ_∞ norm and the ℓ_2 norm. In particular, algorithms for HSVP_2 play a very important role in lattice-based cryptography, and algorithms for HSVP_2 with different approximation factors are a very active area of research [27, 38, 15, 31, 1]. Some of these algorithms can be viewed as constructive proofs of classical results about the minimum distance of a lattice relative to the determinant. (For example, the celebrated LLL algorithm gives a constructive proof of Hermite's bound [27], and the slide reduction algorithm gives a constructive proof of Mordell's inequality [15].)

Finding a vector within Minkowski's bound in the ℓ_∞ norm is considered one of the most important problems in the complexity class PPP. In particular, Ban, Jain, Papadimitriou, Psomas, and Rubinstein showed that other important problems in PPP can be reduced to this problem, and they conjectured that this problem is actually PPP-complete [2]. Sotiraki, Zampetakis, and Zirdelis further investigated the relationship between lattice problems, PPP, and PWPP, showing two problems related to lattices that are PPP-complete and PWPP-complete respectively [41].

The complexity of total coding problems

To our knowledge, much less is known about the complexity of total problems that arise in coding theory. Instead, much work has focused on the *γ-approximate Minimum Distance Problem* (γ-MDP), in which the input is a linear code and the goal is to output a pair of distinct codewords $\boldsymbol{c}_1, \boldsymbol{c}_2$ such that the distance between them is within a factor γ of the minimum distance of the input code (or, since the code is linear, one can equivalently output a non-zero codeword with nearly minimal Hamming weight). This problem is known to be

NP-hard [42], even to approximate [14]. In contrast, we are interested in the problem of finding distinct codewords $\boldsymbol{c}_1, \boldsymbol{c}_2$ that are within distance d, where d depends only on the message length k and block size n of the code (and *not* on the minimum distance). We are particularly interested in the total regime, where such problems are very unlikely to be NP-hard.

To our knowledge, the only direct work in this area is a beautiful paper by Debris-Alazard, Ducas, and van Woerden [11], which showed an LLL-like algorithm for linear codes that efficiently finds codewords within the Griesmer bound [16]. (Note that this algorithm only works for linear codes, and indeed the Griesmer bound itself only applies to linear codes.)

Literature on multicollisions

There is quite a bit of prior work in the cryptography literature on *multicollision-resistant hash functions*. In our terminology, a multicollision-resistant hash function is some efficiently sampleable distribution of instances of (A, B)-Pigeon^L (i.e., circuits $\mathcal{C} : [A] \to [B]$) that are actually hard (i.e., that cannot be solved by polynomial-time algorithms with non-negligible probability). A survey of this literature is beyond the scope of this work. Broadly speaking, work in this area has been concerned with (1) understanding the relationship between collision resistance and multicollision resistance (i.e., the relationship between the case when $L = 2$ and the case when $L > 2$); (2) finding applications of multicollision-resistant hash functions; and (3) building multicollision-resistant hash functions from various cryptographic hardness assumptions.

Many of the techniques that we use to understand the relationship between (A, B)-Pigeon^L in different parameter regimes are directly inspired by this cryptographic literature. In particular, our inclusion based on Merkle trees and list-recoverable codes is a direct adaption of Bitanski, Kalai, and Paneth's construction from the cryptographic setting to our setting [6]; our black-box separations are inspired by [25]; and our non-black-box non-separation is a direct adaptation of Rothblum and Vasudevan's proof from the cryptographic setting to our setting [37]. (See also the very recent work of [9].)

In contrast, there is very little prior work in the worst-case setting. To our knowledge, the only works that consider the worst-case complexity of finding multicollisions are Komargodski, Naor, and Yogev [25] and Sotiraki [40] (though see Section 1.3). In both of these works, the worst-case complexity of (A, B)-Pigeon^L is not the primary focus, but both do define the special cases of the complexity class (A, B)-PMPP^L, when $A = B^2$ (specifically, $A = 2^{2n}$ and $B = 2^n$) and L is a constant. This is a natural setting of parameters, but we show interesting results in other parameter regimes as well.

Sotiraki in particular shows a complete problem for PMPP that is similar to the PPP-complete and PWPP-complete problems from [41]. In particular, Sotiraki's PMPP-complete problem bears some resemblance to certain lattice and coding problems, though the relationship is unclear. We refer the reader to [41, 40] for more.

Komargodski, Naor, and Yogev claimed a black-box separation between $(2^{2n}, 2^n)$-PMPP^L and $(2^{2n}, 2^n)$-PMPP$^{L'}$ for any constants $L' < L$, but their elegant proof contained a subtle bug that has not been fixed [26]. Our black-box separations use similar ideas but are weaker than what they originally claimed.

Literature on PPP and PWPP

In contrast, there is much literature studying the complexity classes PPP and PWPP, which correspond to the special case of PMPP when $L = 2$ (in different parameter regimes). PPP was introduced by Papadimitriou in his seminal work [34]. (PWPP appeared in many works

but seems to have been first named PWPP in [22].) Since then, many problems of interest have been shown to be contained in either PWPP or PPP [22, 2, 41, 8]. Until recently, only a small handful of problems were known to be complete for PPP or PWPP [34, 41, 2]. However, Bourneuf, Folwarczný, Hubáček, Rosen, and Schwartzbach recently showed a number of problems arising from extremal combinatorics that are complete for either PPP or PWPP [8].

There has also been some literature concerned with generalizing PPP and PWPP to classes other than PMPP. In particular, Pasarkar, Papadimitriou, and Yannakakis [35] recently introduced the class Polynomial Long Choice (PLC), which can be thought of as corresponding to the generalization of the pigeonhole principle obtained by iterating the pigeonhole principle many times.

1.5 A note on "codes" represented by circuits, injectivity, and systematic form

The coding theory results in this paper are concerned with the problem of finding close codewords (or many codewords that lie in a relatively small ball) in a "code" *represented by an arbitrary circuit* $\mathcal{C} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ with size $\text{poly}(n, k, \log q)$. It is far more common in the literature to consider *linear* codes represented by a generator matrix (or, equivalently, an invertible circuit with linear gates). (Sometimes when arbitrary codes are considered in the literature, the code is simply represented by listing all q^k points, while we represent our codes succinctly.)

Whether one should really think of a generic circuit $\mathcal{C} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ as representing a "code" is not so clear. In particular, such a circuit might not be *injective*, i.e., there might be distinct "messages" $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{F}_q^k$ that map to the same "codeword" $\mathcal{C}(\boldsymbol{x}_1) = \mathcal{C}(\boldsymbol{x}_2)$. (And, there is likely no way to efficiently determine whether such a circuit is injective. Indeed, determining this is coNP-complete.) But, we find the coding-theoretic perspective to be quite useful. In particular, the notion of the distance of a code still makes sense with this slightly more general definition, and the bounds on the distance that we discuss in this paper still apply. Indeed, much of coding theory still makes sense if we treat such degenerate, non-injective codes simply as "codes with distance 0," and standard bounds in coding theory, such as the Singleton and Hamming bounds, still apply.

Of course, it is not an issue, and actually a strength that our upper bounds apply to such general "codes." Indeed, any upper bound that applies in the more general case when "codes" are represented by arbitrary circuits certainly applies to the special case of injective circuits or the even more special (and quite important) case of linear codes.

For our lower bounds, however, this can be viewed as a major flaw in our model. To partially mitigate this, we prove our hardness results in two different settings.

In the first "generic" setting, we show hardness for "codes" represented by arbitrary circuits $\mathcal{C} : [q]^k \to [q]^n$, with no presumed structure. In particular, the circuit is not necessarily injective. Indeed, one may argue that our reductions are rather artificial, in that the reductions often produce "codes" $\mathcal{C}$ such that $\Delta(\mathcal{C}(\boldsymbol{x}_i), \mathcal{C}(\boldsymbol{x}_j))$ is either zero or strictly larger than d, where d is the bound on the distance needed to solve the associated coding problem. So, these reductions rely quite heavily on this rather strange definition of a code in which multiple messages can map to the same codeword.

In the "systematic" setting, our codes $\mathcal{C} : [q]^k \to [q]^n$ are in *systematic* form, which means that the first k characters of $\mathcal{C}(\boldsymbol{x})$ are simply $\boldsymbol{x}$ itself. (Formally, we actually work with a smaller circuit $\mathcal{C}' : [q]^k \to [q]^{n-k}$, and we simply interpret $(\boldsymbol{x}, \mathcal{C}'(\boldsymbol{x}))$ as the codeword associated with $\boldsymbol{x}$.) Such circuits are clearly injective, and therefore this setting is much less artificial. (Codes in systematic form are also quite fundamental objects that arise naturally in coding theory.) In this setting, our formal results are a bit different, and we even show

that there are efficient algorithms for this setting in regimes where the problem is provably hard for codes represented by arbitrary circuits. However, the high-level picture is the same. In particular, we still show completeness in this setting. (See Figure 1.)

1.6 Open problems

We leave a number of interesting open problems. Here, we mention some of them.

Complexity of coding and lattice problems

We place the computational problems associated with a number of fundamental bounds on the minimum distance of a code in PMPP. However, we were unable to say anything non-trivial about the complexity of Delsarte's linear programming bound [12] and the closely related MRRW bound(s) [29], which are the best known bounds in an important range of parameters. The associated computational problems are, by definition, in TFNP, but we do not know any natural subclass of TFNP that contains these problems.

Similarly, the best known bound on the minimum distance in the ℓ_2 norm of a lattice relative to the determinant is the celebrated bound due to Kabatjanskiĭ and Levenšteĭn [24] (which is better than Blichfeldt's [7] by a factor of roughly $2^{1/10}$). The problem of finding a vector within the Kabatjanskiĭ and Levenšteĭn bound is again, by definition, in TFNP, but we do not know any natural subclass of TFNP that contains this problem.

In a different direction, we show hardness of various total coding problems (and even completeness in some regimes), but only for codes represented by circuits. It would be very exciting to show hardness results of total problems on *linear* codes (represented, e.g., by generator matrices), since these are the more standard problems. (The analogous question for lattices was already asked in [2] and also remains open.)

Better understanding of PMPP across different regimes

The first question that comes to mind about the complexity classes (A, B)-PMPP^L is, of course, "how do these classes relate to each other across different parameter regimes?" In the case when $L = 2$, it has long been known that the relationship between A and B does not matter much. For example,

$$((1 + 1/\operatorname{poly}(n)) \cdot 2^n, 2^n)\text{-PMPP}^2 = ((2^{n+\operatorname{poly}(n)}, 2^n)\text{-PMPP}^2 = \text{PWPP} .$$

For $L > 2$, it seems unlikely that a similar result holds. We instead describe a rich and rather complicated set of inclusions (and non-black-box relationships) among these classes, as well as black-box separations.

However, we have no evidence that these results are tight. One would ideally like to show black-box separations and inclusions that are tight with one another. (Or, alternatively, one might hope to prove non-trivial equalities (A, B)-PMPP$^L = (A', B')$-PMPP$^{L'}$ like what we know in the case of $L = L' = 2$.) The analogous average-case question has been the topic of much research in the cryptography literature [23, 33, 43, 5, 6, 25, 28, 13, 40, 37, 20], but to the authors' knowledge the worst-case setting was barely explored before this work and the recent exciting (concurrent and independent) work of Jain, Li, Robere, and Xun [21].

2 Upper bounds for coding problems

In this section, we prove that SDP and DenseBall are contained in PMPP in many parameter regimes of interest. In particular, for SDP, we show such results when the distance $d(n, k, q)$ corresponds to a number of celebrated bounds on the minimum distance of a code, including

the Singleton bound (Corollary 7), the Plotkin bound (Corollary 8), the Hamming bound (Corollary 13), and the Elias-Bassalygo bound (Corollary 15). For DenseBall, we show analogous results for the list Singleton bound (Corollary 11) and list Hamming bound (Corollary 13).

In fact, all of these results are corollaries of four technical results: Theorem 6, which reduces SDP to Pigeon via a simple truncation technique; Theorem 10, which does something similar for DenseBall; Theorem 12, which reduces DenseBall to Pigeon by "finding many overlapping Hamming balls centered on codewords"; and Theorem 14, which shows a generic reduction from SDP to DenseBall.

We also show in Section 2.1.1 an efficient algorithm for SDP within the Plotkin bound for codes in systematic form.

2.1 Upper bounds for the Singleton and Plotkin bounds

We start by giving a fairly general reduction from SDP (the problem of finding a pair of close codewords) to Pigeon^L. On input an instance $\mathcal{C}$ of SDP, the reduction works by defining a compressing circuit $\mathcal{C}'$ that truncates the output of $\mathcal{C}$. It then finds $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L$ such that $\mathcal{C}'(\boldsymbol{x}_1) = \cdots = \mathcal{C}'(\boldsymbol{x}_L)$ using its Pigeon^L oracle, and finally outputs a pair $\boldsymbol{x}_i \neq \boldsymbol{x}_j$ that minimizes $\Delta(\mathcal{C}(\boldsymbol{x}_i), \mathcal{C}(\boldsymbol{x}_j))$. We then observe that special cases of this general result place (n, k, d)-SDP$_q$ in PWPP for d corresponding to the Singleton bound (Corollary 7) and PMPP for d corresponding to the Plotkin bound (Corollary 8).

▶ **Theorem 6.** *Let $k, m, n, q, L \in \mathbb{Z}^+$ be such that $m \leq k \leq n$ and $2 \leq L \leq q^{k-m}$ with $L \leq \mathrm{poly}(n, \log q)$. Then there is a Karp reduction from (n, k, d)-SDP$_q$ to (q^k, q^m)-Pigeon^L where $d := d_q^*(n - m, L)$.*

Proof. See full version [4]. ◀

We get useful corollaries from Theorem 6 that show how to compute vectors achieving the Singleton bound (the $L = 2$ case) and the Plotkin bound (for larger L) using a Pigeon^L oracle.

We now show that the problem of finding a pair of codewords whose distance is at most the Singleton bound is in PWPP.

▶ **Corollary 7** (Singleton bound in PWPP)**.** *Let $k, n, q \in \mathbb{Z}^+$ with $k \leq n$. Then there is a Karp reduction from $(n, k, n - k + 1)$-SDP$_q$ to (q^k, q^{k-1})-Pigeon^2. In particular, $(n, k, n - k + 1)$-SDP$_q$ is in PWPP.*

Proof. The claim follows from Theorem 6 by setting $m := k - 1$ and noting that, trivially, $d_q^*(n - m, L) = d_q^*(n - k + 1, L) \leq n - k + 1$ for any $L \geq 2$. ◀

We now show that the problem of computing a pair of codewords whose distance satisfies the high-rate Plotkin bound is in PMPP.

▶ **Corollary 8** (Plotkin bound in PMPP)**.** *Let $k, n, d, m, q, L \in \mathbb{Z}^+$ be such that $k \leq n$, $L \leq \mathrm{poly}(n)$, $L \leq q^{k-m}$, and*

$$d > \frac{L}{L-1}\left(1 - \frac{1}{q}\right)(n - m)\ , \tag{1}$$

then there is a Karp reduction from $(n, k, d-1)$-SDP$_q$ to (q^k, q^m)-Pigeon^L which runs in time $\mathrm{poly}(n, \log(q))$.

Proof. See full version [4]. ◀

2.1.1 Efficiently solving sysSDP up to the Plotkin bound

We now note that there is an efficient algorithm for finding codewords within the Plotkin bound *when the code is in systematic form.* In other words, we show that (n, k, d)-sysSDP$_q$ can be solved efficiently for d below the Plotkin bound, where we recall that sysSDP is the special case of SDP in which the code is in systematic form.

▶ **Theorem 9.** *Let $k, n, q \in \mathbb{Z}^+$ with $k < n$ and $q \geq 2$. There is a* poly(n, q)*-time algorithm for* (n, k, d)-sysSDP$_q$ *where*

$$d := (1 - 1/q)(n - k + \lceil \log_q(4qn) \rceil) \ .$$

Proof. See full version [4]. ◀

2.2 An upper bound for the list Singleton bound

Next, we give a reduction from DenseBall to Pigeon similar to the reduction from SDP to Pigeon in Theorem 6. We then show that this reduction places the *list* Singleton bound (i.e., a variant of the Singleton bound that corresponds to list decoding rather than unique decoding) in PMPP (up to a small error).

▶ **Theorem 10.** *Let $k, n, m, q, L \in \mathbb{Z}^+$ be such that $m < n$ and $2 \leq L \leq q^{k-m}$. Then there is a Karp reduction from* $(n, k, n - m - \lfloor (n - m)/L \rfloor)$-DenseBall$_q^L$ *to* (q^k, q^m)-Pigeon^L.

Proof. See full version [4]. ◀

Finally, we show that the problem of computing a set of codewords lying in a ball whose radius *almost* satisfies the list Singleton bound is in PMPP.

▶ **Corollary 11** (List Singleton bound in PMPP)**.** *Let $n, k, q, L \in \mathbb{Z}^+$ with $k > \lfloor \log_q(L) \rfloor$, let $m := k - \lfloor \log_q L \rfloor$, and let*

$$t := n - m + \lfloor (n - m)/L \rfloor = n - k + \lfloor \log_q L \rfloor - \lfloor (n - k + \lfloor \log_q L \rfloor)/L \rfloor \ . \tag{2}$$

Then (n, k, t)-DenseBall$_q^L$ *is in* (q^k, q^m)-PMPP^L.

Proof. See full version [4]. ◀

We remark that the radius t in Equation (2) is almost as small the smallest t satisfying the list Singleton bound. Indeed, one can check that any t satisfying the list Singleton bound must be such that

$$t > (1 - 1/L) \cdot (n - k + \log_q(L - 1)) \ ,$$

and that the radius t in Equation (2) satisfies

$$t \leq (1 - 1/L) \cdot (n - k + \lfloor \log_q(L) \rfloor) + 1 \ .$$

2.3 Upper bounds for the (list) Hamming and Elias-Bassalygo bounds

We now show containment of (n, k, d)-SDP$_q$ in PMPP, where d corresponds to the Hamming bound and Elias-Bassalygo bound. In particular, when d is equal to the value given by the Hamming bound, we show that SDP is contained in PPP (see Item 1 of Corollary 13) and for d slightly above the Hamming bound, SDP is contained in PWPP (see Item 2 of Corollary 13). In fact, we show a more general result that applies to the generalization (n, k, d)-DenseBall$_q^L$

of SDP and shows that for d within the *list* Hamming bound, this problem is in PMPP (see Item 3 of Corollary 13). We then show that (n, k, d)-SDP$_q$ is in PMPP whenever d above the Elias-Bassalygo, via a generic reduction from SDP to DenseBall (see Corollary 15). In fact, we obtain a smooth tradeoff between the distance d obtained by the reduction and the parameters A, B, and L in the reduction to (A, B)-Pigeon^L.

2.3.1 The (list) Hamming bound

We now give a reduction from DenseBall to Pigeon^L, the key to which is the circuit $\mathcal{C}_H^{n,V,q}$ giving an injection from $[V]$ into a Hamming ball for sufficiently large V defined in the full version [4]. (In fact, we will use such circuits $\mathcal{C}_H^{n,V,q}$ that are *bijective*.)

▶ **Theorem 12.** *Let $n, k, t, L \in \mathbb{Z}^+$, and let q be a prime power satisfying*

$$L \leq \left\lceil \frac{q^k \cdot V_q^n(t)}{q^n} \right\rceil .$$

Then there is a Karp reduction from (n, k, t)-DenseBall$_q^L$ to $(q^k \cdot V_q^n(t), q^n)$-Pigeon^L.

Proof. See full version [4]. ◀

We now use Theorem 12 to show that SDP and DenseBall with parameters corresponding to the (list) Hamming bound are in (one or more of) PPP, PWPP, and PMPP.

▶ **Corollary 13** ((List) Hamming bound in PPP, PWPP, and PMPP)**.** *Let $n, k, t, q, L \in \mathbb{Z}^+$ with $2 \leq q \leq \mathrm{poly}(n)$ and $L \leq \mathrm{poly}(n)$. Let*

$$\widetilde{L} := \frac{q^k \cdot V_q^n(t)}{q^n} .$$

Then:
1. *If $\widetilde{L} > 1$, then $(n, k, 2t)$-SDP$_q$ is in PPP.*
2. *If $\widetilde{L} \geq 1 + 1/\mathrm{poly}(n)$, then $(n, k, 2t)$-SDP$_q$ is in PWPP.*
3. *If $\widetilde{L} \geq L$, then (n, k, t)-DenseBall$_q^L$ is in $(q^k \cdot V_q^n(t), q^n)$-PMPP^L.*

Proof. See full version [4]. ◀

2.3.2 E pluribus duo – from many codewords to two

We now give a simple reduction from SDP to DenseBall^L for $L \geq 2$. The idea is that a ball of small radius t containing $L \geq 2$ codewords $\mathcal{C}(\boldsymbol{x}_1), \ldots, \mathcal{C}(\boldsymbol{x}_L)$ must contain a pair of codewords at small distance d. Clearly such a pair must exist at distance at most $2t$, but in fact when L is larger it is possible to get a better upper bound. So, the reduction simply computes the distance $\Delta(\mathcal{C}(\boldsymbol{x}_i), \mathcal{C}(\boldsymbol{x}_j))$ between each pair of codewords $\mathcal{C}(\boldsymbol{x}_i), \mathcal{C}(\boldsymbol{x}_j)$ for $i \neq j$, and outputs a pair $\boldsymbol{x}_i, \boldsymbol{x}_j$ that minimizes $\Delta(\mathcal{C}(\boldsymbol{x}_i), \mathcal{C}(\boldsymbol{x}_j))$.

▶ **Theorem 14.** *Let $n, k, t, q, L \in \mathbb{Z}^+$ be such that*

$$L \leq \left\lceil \frac{q^k \cdot V_q^n(t)}{q^n} \right\rceil .$$

Then there is a $\mathrm{poly}(n, L, \log q)$-time Karp reduction from (n, k, d)-SDP$_q$ to (n, k, t)-DenseBall$_q^L$, where $d := d_{[q]^n, \Delta}(t, L)$.

Proof. See full version [4]. ◀

2.3.3 The Elias-Bassalygo bound

We are now ready to show that the problem of computing a pair of codewords satisfying the Elias-Bassalygo bound is in PMPP.

▶ **Corollary 15** (Elias-Bassalygo bound in PMPP). *Let $n, k, d, t, q \in \mathbb{Z}^+$ with $t < J_q(d/n) \cdot n$, let $L := qnd + 1$, and suppose that*

$$L \leq \left\lceil \frac{q^k \cdot V_q^n(t)}{q^n} \right\rceil .$$

Then there is a poly(n, q)*-time Karp reduction from $(n, k, d-1)$-SDP$_q$ to $(q^k \cdot V_q^n(t), q^n)$-*Pigeon^L*. In particular, $(n, k, d-1)$-SDP$_q$ is in $(q^k \cdot V_q^n(t), q^n)$-*PMPP^L*.*

Proof. See full version [4]. ◀

3 Hardness of coding problems

We now turn to proving *hardness results* for SDP and DenseBall. In each of our hardness reductions we first assume that a gadget code, specified by a circuit $\mathcal{C}_A$, is given to the reduction as auxiliary input. We then instantiate the gadget code $\mathcal{C}_A$ with an explicit efficiently computable code $\mathcal{C}_A$ to obtain a true Karp reduction.

3.1 PWPP-hardness of SDP

We first show a generic reduction, with auxiliary from Pigeon^2 to SDP.

▶ **Theorem 16.** *Let $k, m, n, d, d' \in \mathbb{Z}^+$ be such that $m < k$ and $d < d'$. Let q be a prime power. Then there is a Karp reduction from (q^k, q^m)-*Pigeon^2 *to (n, k, d)-SDP$_q$ that takes a circuit $\mathcal{C}_A : \mathbb{F}_q^m \to \mathbb{F}_q^n$ defining an $(n, m, d')_q$ code as auxiliary input.*

Proof. See full version [4]. ◀

We will instantiate the reduction in Theorem 16 with codes $\mathcal{C}_A$ meeting the Zyablov bound, which can be computed in polynomial time (see [18, Theorem 10.2.1]). In fact, what we state is the special case of the (effective) Zyablov bound with sub-constant rate.[7]

▶ **Theorem 17** (Zyablov bound [44]). *For any (efficiently computable) $k = k(n) = o(n)$, constant prime power q, and constant $\varepsilon > 0$, there is a* poly(n)*-time algorithm that takes as input n and computes (a circuit representing) a code $\mathcal{C}_q^{\mathsf{Zyb}} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ with relative distance $\delta \geq 1 - 1/q - \varepsilon$ for sufficiently large n.*

We now show that SDP is PWPP-hard on q-ary codes of relative distance $\delta := d/n$ less than $1 - 1/q$ and any constant rate. This hardness corresponds to the red shaded region in the top plot in Figure 1. We remark that $(n, k, \delta n)$-SDP$_q$ is easy for $\delta \geq 1 - 1/q$. In particular, for such any such δ, this problem can be solved by choosing any collection of polynomially many codewords and outputting the closest pair among them. So, Corollary 18 shows essentially tight hardness of SDP in terms of δ.

[7] We note that the premise of [18, Theorem 10.2.1] is stated with the requirement $\delta < 1/2$. While $\delta < 1/2$ is necessary for the $q = 2$ case, for general constant q, the result holds for $\delta < 1 - 1/q$, as in Theorem 17. It is also evident from the proof of the Zyablov bound that the result extends to superconstant q as well – i.e., that for any prime power q that is an unbounded function of n and any $k = o(n)$, there is an efficiently computable family of codes $\mathcal{C} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ with relative distance $1 - \varepsilon$.

▶ **Corollary 18** (PWPP-hardness of SDP). *Let q be a fixed prime power and let $\varepsilon, \varepsilon' > 0$ be constants. Then for all sufficiently large $n, k, d \in \mathbb{Z}^+$ with $k \le n \le \mathrm{poly}(k)$ and $d \le (1 - 1/q - \varepsilon')n$, there is a Karp reduction from $(q^k, q^{k^{1-\varepsilon}})$-Pigeon^2 to (n, k, d)-SDP$_q$.*

In particular, $(n, Rn, \delta n)$-SDP$_q$ is PWPP*-hard for any constant rate for any constant rate $R \in (0, 1]$ and constant relative distance $\delta \in (0, 1 - 1/q)$.*

Proof. See full version [4]. ◀

▶ Remark 19. We remark that the assumption $k \le n$ (equivalently, $R \le 1$) in Corollary 18 is not necessary, except to ensure that the instance $\mathcal{C}'$ of SDP output by the reduction meets the definition of a code used there. In fact, modifying Corollary 18 slightly shows PWPP-hardness of "(n, k, d)-SDP$_q$" for any $n \ge \Omega(k^\varepsilon)$ for constant $\varepsilon > 0$ (and any constant relative distance less than). This is because the hard instances $\mathcal{C}'$ of SDP constructed in Corollary 18 are such that either $\mathcal{C}'(\boldsymbol{x}) = \mathcal{C}'(\boldsymbol{y})$ or $\Delta(\mathcal{C}'(\boldsymbol{x}), \mathcal{C}'(\boldsymbol{y}))$ is large for $\boldsymbol{x} \neq \boldsymbol{y}$.

We also extend Corollary 18 to codes in systematic form (and in particular to codes with distance $d \ge 1$, which are represented by injective circuits; see Section 1.5). This hardness corresponds to the red shaded region in the bottom plot in Figure 1.

▶ **Corollary 20** (PWPP-hardness of sysSDP). *Let q be a fixed prime power and let $\varepsilon, \varepsilon' > 0$ be constants. Then for all sufficiently large $n, k, d \in \mathbb{Z}^+$ with $k \le n - k \le \mathrm{poly}(k)$ and $d \le (1 - 1/q - \varepsilon')(n - k)$, there is a Karp reduction from $(q^k, q^{(n-k)^{1-\varepsilon}})$-Pigeon^2 to (n, k, d)-sysSDP$_q$.*

In particular, $(n, Rn, \delta n)$-sysSDP$_q$ is PWPP*-hard for any constant rate $R \in (0, 1)$ and constant relative distance $\delta \in (0, (1 - 1/q) \cdot (1 - R))$.*

Proof. See full version [4]. ◀

3.2 PMPP-hardness of DenseBall

We next show a reduction from Pigeon^L for $L \ge 2$ to DenseBall analogous to Theorem 16.

▶ **Theorem 21.** *Let $k, m, n, t, L, L_A, L' \in \mathbb{Z}^+$ and let q be a prime power. Suppose that $L' \le \min\{\mathrm{poly}(n), q^{n-k}\}$ and $L \le \min\{\lceil L'/L_A \rceil, q^{k-m}\}$. Then there is a Karp reduction from (q^k, q^m)-Pigeon^L to (n, k, t)-DenseBall$_q^{L'}$ that takes a circuit $\mathcal{C}_A : \mathbb{F}_q^m \to \mathbb{F}_q^n$ defining a (t, L_A)-list decodable-code with minimum distance at least 1 (i.e., $\mathcal{C}_A$ is injective) as auxiliary input.*

Proof. See full version [4]. ◀

We now state a result on explicit codes that nearly achieve list-decoding capacity [17] (see also [18, Theorem 17.3.8]). We note in passing that these codes are folded Reed-Solomon codes, but we will only use them in a black-box way as our gadget codes $\mathcal{C}_A$ in Theorem 21.

▶ **Theorem 22** ([17], [18, Theorem 17.3.8]). *For any constant rate $R^* \in (0, 1)$, any sufficiently small constant $\varepsilon > 0$, and all sufficiently large $n \in \mathbb{Z}^+$, there exist linear q-ary $((1 - R^* - \varepsilon)n, L_A)$-list-decodable codes $\mathcal{C}_q^{\mathsf{FRS}}$ of dimension $R^* \cdot n$ and sufficiently large block length n, for some*

$$L_A \le \left(\frac{n}{\varepsilon}\right)^{O(1/\varepsilon)}, \quad q \le \left(\frac{n}{\varepsilon}\right)^{O(1/\varepsilon^2)}.$$

Furthermore, there is a $\mathrm{poly}(n)$-time algorithm for computing (a circuit representing) such codes $\mathcal{C}_q^{\mathsf{FRS}}$.

Finally, we use Theorem 22 to prove PMPP-hardness of DenseBall.

▶ **Corollary 23** (PMPP-hardness of DenseBall)**.** *For any constants $R, \rho \in (0,1)$ and positive integers $m := m(n) < o(n)$ and $L := L(n) \leq \text{poly}(n)$, there exists a prime power $q := q(n) \leq \text{poly}(n)$ and a list size $L' := L'(n) \leq \text{poly}(n)$ such that there is a Karp reduction from (q^k, q^m)-Pigeon^L to $(n, k, \rho n)$-DenseBall$_q^{L'}$, where $k := k(n) = \lfloor Rn \rfloor$.*

In particular, for these parameters, $(n, k, \rho n)$-DenseBall$_q^{L'}$ is (q^k, q^m)-PMPP^L-hard.

Proof. See full version [4]. ◀

As with SDP, we also show hardness of DenseBall for codes in systematic form. (See Section 1.5.)

▶ **Corollary 24** (PMPP-hardness of sysDenseBall)**.** *For any constants $R \in (0,1)$ and $0 < \rho < 1 - R$ and positive integers $m := m(n) < o(n)$ and $L := L(n) \leq \text{poly}(n)$, there exists a prime power $q = q(n) \leq \text{poly}(n)$ and a list size $L' = L'(n) \leq \text{poly}(n)$ such that there is a Karp reduction from (q^k, q^m)-Pigeon^L to $(n, k, \rho n)$-sysDenseBall$_q^{L'}$, where $k := k(n) = \lfloor Rn \rfloor$.*

In particular, for these parameters, $(n, k, \rho n)$-sysDenseBall$_q^{L'}$ is (q^k, q^m)-PMPP^L-hard

Proof. See full version [4]. ◀

4 Finding short lattice vectors is in PMPP

In this section, we show that the problem of finding suitably short non-zero lattice vectors (in ℓ_p norms) can be placed in (A, B)-PMPP^L, with a smooth tradeoff between the length of the vector obtained and L and B. In particular, when $L = 2$ and $A \approx B$, we find vectors whose length is at most the bound given by Minkowski's celebrated theorem [32] (up to a factor of $1 + 1/n^C$ for an arbitrarily large constant $C > 0$), and when $L = \text{poly}(n)$ and $A \approx LB$, we find shorter vectors, corresponding to a stronger bound due to Blichfeldt [7]. (Blichfeldt proved his bound in the ℓ_2 norm, but we generalize it. Again, we match Blichfeldt's bound up to a factor of $1 + 1/n^C$ for an arbitrarily large constant $C > 0$.)

To that end, we first prove the following technical proposition. We then derive the above results as corollaries.

▶ **Proposition 25.** *There is an algorithm that takes as input $p \geq 1$, a radius $r \geq 1$, an integer $q \geq 1$, integers $A > B \geq 1$, an integer $L \geq 1$, and a basis $\mathbf{B} \in \mathbb{Z}^{n \times n}$ for a lattice $\mathcal{L} = \mathcal{L}(\mathbf{B})$, makes a single query to an (A, B)-Pigeon^L oracle, and outputs a lattice vector $\boldsymbol{y} \in \mathcal{L}(\mathbf{B})$ with the following behavior. If $L \leq \lceil A/B \rceil$, $q \geq 100pn/r$ and*

$$q^n \det(\mathcal{L}) \leq B < A \leq \left(1 - \frac{20n}{rq}\right) \cdot q^n \cdot \text{vol}(\mathcal{B}_p^n(r)) \,,$$

then $0 < \|\boldsymbol{y}\|_p \leq d_{\ell_p^n}(r, L)$. Furthermore, the algorithm runs in time $\text{poly}(n, L, \log A, \log r, \log \|\mathbf{B}\|)$.

Proof. See full version [4] ◀

Recall that we are interested in γ-HSVP$_p$ for $\gamma \approx \text{vol}(\mathcal{B}_p^n(1))^{-1/n}$.

▶ **Corollary 26.** *Let $p \geq 1$ be a constant integer, and let $A = A(n), B = B(n), L = L(n) \in \mathbb{Z}^+$ be such that $2^{s(n)} = B < A \leq 2^{\text{poly}(n)}$ and $L \leq \lceil A/B \rceil$ for a sufficiently large polynomial $s(n)$. Then γ-HSVP$_p \in (A, B)$-Pigeon^L for*

$$\gamma := (1 + o(1)) \cdot \text{vol}(\mathcal{B}_p^n(1))^{-1/n} \cdot (A/B)^{1/n} \cdot d_{\ell_p^n}(1, L) \,,$$

where n is the rank of the lattice $\mathcal{L}$ in the γ-HSVP$_p$ instance.

In particular, $\gamma_M := 2 \cdot \mathrm{vol}(\mathcal{B}_p^n(1))^{-1/n}$ *corresponds to Minkowski's theorem , and* $((1 + o(1))\gamma_M)$-$\mathrm{HSVP}_p \in \mathsf{PWPP}$. *Furthermore, for the case of* $p = 2$, $\gamma_B := \sqrt{2} \cdot \mathrm{vol}(\mathcal{B}_2^n(1))^{-1/n}$ *corresponds to Blichfeldt's theorem , and* $((1 + o(1))\gamma_B)$-$\mathrm{HSVP}_2 \in (2LB, B)$-$\mathsf{PMPP}^L$ *for any* $L = \mathrm{poly}(\log B)$.

Proof. See full version [4]. ◀

5 Inclusions

See full version [4].

6 Black-box separations

See full version [4].

7 A non-black-box non-separation

See full version [4].

References

1 Divesh Aggarwal, Zeyong Li, and Noah Stephens-Davidowitz. A $2^{n/2}$-time algorithm for $\sqrt{n}$-SVP and $\sqrt{n}$-Hermite SVP, and an improved time-approximation tradeoff for (H)SVP. In *Eurocrypt*, 2021. URL: `http://arxiv.org/abs/2007.09556`.

2 Frank Ban, Kamal Jain, Christos H. Papadimitriou, Christos-Alexandros Psomas, and Aviad Rubinstein. Reductions in PPP. *Information Processing Letters*, 145:48–52, 2019. `doi:10.1016/j.ipl.2018.12.009`.

3 L. A. Bassalygo. New upper bounds for error-correcting codes. *Problems of Information Transmission*, pages 32–35, 1965.

4 Huck Bennett, Surendra Ghentiyala, and Noah Stephens-Davidowitz. The more the merrier! on total coding and lattice problems and the complexity of finding multicollisions. In *ITCS*, 2025. URL: `https://eccc.weizmann.ac.il/report/2024/018`.

5 Itay Berman, Akshay Degwekar, Ron D. Rothblum, and Prashant Nalini Vasudevan. Multi-collision resistant hash functions and their applications. In *Eurocrypt*, 2018.

6 Nir Bitansky, Yael Tauman Kalai, and Omer Paneth. Multi-collision resistance: A paradigm for keyless hash functions. In *STOC*, 2018.

7 H. F. Blichfeldt. The minimum value of quadratic forms, and the closest packing of spheres. *Mathematische Annalen*, 101(1):605–608, 1929.

8 Romain Bourneuf, Lukáš Folwarczný, Pavel Hubáček, Alon Rosen, and Nikolaj I. Schwartzbach. PPP-completeness and extremal combinatorics. In *ITCS*, 2023.

9 Jan Buzek and Stefano Tessaro. Collision resistance from multi-collision resistance for all constant parameters. In *CRYPTO*, 2024.

10 Ivan Bjerre Damgård. A design principle for hash functions. In *CRYPTO*, 1989.

11 Thomas Debris-Alazard, Léo Ducas, and Wessel P. J. van Woerden. An algorithmic reduction theory for binary codes: LLL and more. *IEEE Transactions on Information Theory*, 68(5):3426–3444, 2022. `doi:10.1109/TIT.2022.3143620`.

12 Philippe Delsarte. *An algebraic approach to the association schemes of coding theory.* Thesis, Universite Catholique de Louvain, 1973.

13 Itai Dinur. Tight time-space lower bounds for finding multiple collision pairs and their applications. In *Eurocrypt*, 2020.

14 I. Dumer, D. Micciancio, and M. Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Transactions on Information Theory*, 49(1):22–37, 2003. doi:10.1109/TIT.2002.806118.

15 Nicolas Gama and Phong Q. Nguyen. Finding short lattice vectors within Mordell's inequality. In *STOC*, 2008.

16 J. H. Griesmer. A bound for error-correcting codes. *IBM Journal of Research and Development*, 4(5):532–542, 1960. doi:10.1147/RD.45.0532.

17 Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Trans. Inf. Theory*, 54(1):135–150, 2008. doi:10.1109/TIT.2007.911222.

18 Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. *Essential Coding Theory*. self-published, 2023. October 3rd, 2023 book version. URL: https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/web-coding-book.pdf.

19 R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, 1950. doi:10.1002/j.1538-7305.1950.tb00463.x.

20 Yassine Hamoudi and Frédéric Magniez. Quantum time–space tradeoff for finding multiple collision pairs. *ACM Trans. Comput. Theory*, 15(1-2):3:1–3:22, 2023. doi:10.1145/3589986.

21 Siddhartha Jain, Jiawei Li, Robert Robere, and Zhiyang Xun. On pigeonhole principles and Ramsey in TFNP. In *FOCS*, 2024.

22 Emil Jeřábek. Integer factoring and modular square roots. *Journal of Computer and System Sciences*, 82(2):380–394, 2016. doi:10.1016/J.JCSS.2015.08.001.

23 Antoine Joux. Multicollisions in iterated hash functions. application to cascaded constructions. In *CRYPTO*, 2004.

24 Grigorii A. Kabatjanskiĭ and Vladimir I. Levenšteĭn. Bounds for packings on the sphere and in space. *Problemy Peredači Informacii*, 14(1):3–25, 1978.

25 Ilan Komargodski, Moni Naor, and Eylon Yogev. Collision resistant hashing for paranoids: Dealing with multiple collisions. In *Eurocrypt*, 2018.

26 Ilan Komargodski and Eylon Yogev. Personal communication, 2023.

27 Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, December 1982.

28 Qipeng Liu and Mark Zhandry. On finding quantum multi-collisions. In *Eurocrypt*, 2019.

29 R. McEliece, E. Rodemich, H. Rumsey, and L. Welch. New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities. *IEEE Transactions on Information Theory*, 23(2):157–166, 1977. doi:10.1109/TIT.1977.1055688.

30 Ralph C. Merkle. A certified digital signature. In *CRYPTO*, 1989.

31 Daniele Micciancio and Michael Walter. Practical, predictable lattice basis reduction. In *Eurocrypt*, 2016. URL: http://eprint.iacr.org/2015/1123.

32 Hermann Minkowski. *Geometrie der Zahlen*. B.G. Teubner, 1910. URL: http://books.google.com/books?id=MusGAAAAYAAJ.

33 Mridul Nandi and Douglas R. Stinson. Multicollision attacks on some generalized sequential hash functions. *IEEE Transactions on Information Theory*, 53(2):759–767, 2007. doi:10.1109/TIT.2006.889721.

34 Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.*, 48(3):498–532, 1994. doi:10.1016/S0022-0000(05)80063-7.

35 Amol Pasarkar, Christos Papadimitriou, and Mihalis Yannakakis. Extremal combinatorics, iterated pigeonhole arguments and generalizations of PPP. In *ITCS*, 2023.

36 M. Plotkin. Binary codes with specified minimum distance. *IRE Transactions on Information Theory*, 6(4):445–450, 1960. doi:10.1109/TIT.1960.1057584.

37 Ron D. Rothblum and Prashant Nalini Vasudevan. Collision-resistance from multi-collision-resistance. In *CRYPTO*, 2022.

38 Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53(23):201–224, 1987. doi:10.1016/0304-3975(87)90064-8.

39 R. Singleton. Maximum distance q-nary codes. *IEEE Transactions on Information Theory*, 10(2):116–118, 1964. doi:10.1109/TIT.1964.1053661.

40 Aikaterini Sotiraki. *New Hardness Results for Total Search Problems and Non-Interactive Lattice-Based Protocols*. Thesis, Massachusetts Institute of Technology, 2020. URL: https://dspace.mit.edu/handle/1721.1/129310.

41 Katerina Sotiraki, Manolis Zampetakis, and Giorgos Zirdelis. PPP-completeness with connections to cryptography. In *FOCS*, 2018.

42 Alexander Vardy. Algorithmic complexity in coding theory and the Minimum Distance Problem. In *STOC*, 1997.

43 Hongbo Yu and Xiaoyun Wang. Multi-collision attack on the compression functions of MD4 and 3-pass HAVAL. In *ICISC*, 2007.

44 Victor Zyablov. An estimate of the complexity of constructing binary linear cascade codes. *Probl. Peredachi Inf.*, 1971.

Exponential-Time Approximation (Schemes) for Vertex-Ordering Problems

Matthias Bentert
University of Bergen, Norway

Fedor V. Fomin
University of Bergen, Norway

Tanmay Inamdar
Indian Institute of Technology Jodhpur, India

Saket Saurabh
The Institute of Mathematical Sciences, Chennai, India
University of Bergen, Norway

Abstract

In this paper, we begin the exploration of vertex-ordering problems through the lens of exponential-time approximation algorithms. In particular, we ask the following question: Can we simultaneously beat the running times of the fastest known (exponential-time) exact algorithms and the best known approximation factors that can be achieved in polynomial time? Following the recent research initiated by Esmer et al. (ESA 2022, IPEC 2023, SODA 2024) on vertex-subset problems, and by Inamdar et al. (ITCS 2024) on graph-partitioning problems, we focus on vertex-ordering problems. In particular, we give positive results for Feedback Arc Set, Optimal Linear Arrangement, Cutwidth, and Pathwidth. Most of our algorithms build upon a novel "balanced-cut" approach – which is our main conceptual contribution. This allows us to solve various problems in very general settings allowing for directed and arc-weighted input graphs. Our main technical contribution is a $(1+\varepsilon)$-approximation for any $\varepsilon > 0$ for (weighted) Feedback Arc Set in $O^*((2-\delta_\varepsilon)^n)$ time, where $\delta_\varepsilon > 0$ is a constant only depending on ε.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis; Mathematics of computing → Graph algorithms

Keywords and phrases Feedback Arc Set, Cutwidth, Optimal Linear Arrangement, Pathwidth

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.15

Funding *Matthias Bentert*: Supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819416).
Fedor V. Fomin: Supported by the Research Council of Norway via the project BWCA (grant no. 314528).
Tanmay Inamdar: Supported by IITJ Research Initiation Grant (grant number I/RIG/TNI/20240072).
Saket Saurabh: Supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819416); and he also acknowledges the support of Swarnajayanti Fellowship grant DST/SJF/MSA-01/2017-18.

1 Introduction

Many NP-hard problems admit algorithms running in time $O^*(2^n)$[1]. Moreover, for fundamental problems like Satisfiability or Set Cover, there are plausible conjectures, namely Strong Exponential Time Hypothesis (SETH) or the Set Cover Conjecture, respectively,

[1] Throughout the introduction, n will have the "natural" meaning, e.g., number of variables/vertices/elements, and the O^*-notation hides polynomial factors. See Section 2 for more details about the latter.


© Matthias Bentert, Fedor V. Fomin, Tanmay Inamdar, and Saket Saurabh;
licensed under Creative Commons License CC-BY 4.0



which state that these problems do not admit algorithms running in time $O^*((2-\varepsilon)^n)$ for any $\varepsilon > 0$. Although these conjectures have been used as a source of analogous running-time lower bound for other problems, many graph problems are not covered by such results. Indeed, quite a few long-standing exponential-time barriers have finally been overcome in recent years [33, 37, 34], giving new life to the field of *moderately exponential-time algorithms* [22].

Improving upon the trivial $O^*(2^n)$-time algorithm for "vertex-subset problems" on graphs, Fomin et al. [21] designed a general framework called *monotone local search* (MLS). This framework enables the creation of state-of-the-art exact exponential algorithms for a wide range of such problems by leveraging the corresponding parameterized algorithms as black-box components. On the other hand, for graph partitioning and vertex-ordering problems, even the $O^*(2^n)$-time algorithm does not follow from a trivial brute force of the search space. Nevertheless, many such problems do admit $O^*(2^n)$-time algorithms via advanced techniques such as Held-Karp-style dynamic programming [23, 6] or speeding up the natural dynamic program via fast subset convolution [26]. For these latter two classes of problems, beating the $O^*(2^n)$ barrier has proven to be challenging, with the notable exception of k-Cut for which Lokshtanov et al. [31] designed an $O^*((2-\varepsilon)^n)$-time algorithm for some fixed $\varepsilon > 0$. Due to the lack of any progress for many of these intensively studied vertex-ordering problems, we think it is the right time to relax the question just slightly in the following way.

> Can we show or exclude $(1+\varepsilon)$-approximations for vertex-ordering problems that run in $O((2-\delta)^n)$ time?

We show that this question can be answered affirmatively for Feedback Arc Set, but we were not able to achieve the same for any other problem. However, the methods we develop for Feedback Arc Set allow us to answer another natural (weaker) relaxation:

> Can we simultaneously beat the running times of the fastest known (exponential-time) exact algorithms and the best known approximation factors that can be achieved in polynomial-time for vertex-ordering problems?

Given the difficulty in surmounting the current-best exact exponential-time algorithms, there has been a growing research interest in answering the latter relaxation. This is also justified given the fact that most of the problems are known to be APX-hard – they do not admit arbitrarily good approximations. In some cases, problems do not even admit constant-factor approximations in polynomial time assuming $P \neq NP$. To the best of our knowledge, the first exponential-time approximation algorithms were given by Williams [36] who showed how to generalize $O^*(2^{\omega n/3})$-time algorithms for (unweighted) Max Cut and Max 2-SAT to $(1+\varepsilon)$-approximations for the weighted variants using "rounding tricks"; here ω denotes the fast matrix multiplication constant. Quite recently, Alman et al. [1] showed that Max Sat can also be approximated faster than $O^*(2^n)$, that is, they showed that for each $\varepsilon > 0$, there is a $(1+\varepsilon)$-approximation in $O^*((2-\delta_\varepsilon)^n)$ time where $\delta_\varepsilon > 0$ is a constant only depending on ε. Moreover, the monotone local search framework of Fomin et al. [21] was extended to give exponential-time approximation schemes for vertex-subset problems in the works of Esmer et al. [15, 16, 17]. Even more recently, Inamdar et al. [26] gave a framework for designing exponential-time approximation schemes for graph partitioning (i.e., cut) problems and a few more examples for specific problems are known [8, 9, 4]. However, none of these techniques seem applicable to vertex-ordering problems, where the objective is to find an ordering of vertices that satisfies some polynomial-time satisfiable predicate, such as Feedback Arc Set, Optimal Linear Arrangement, Cutwidth, and Pathwidth. For all of the above except Pathwidth, the $O^*(2^n)$-time dynamic programming remains the

Table 1 An overview of our results for unweighted input graphs. For each problem in the left column, we show an algorithm with approximation factor in the middle column and the running time is shown in the right column. The value $\omega < 2.373$ is the fast matrix multiplication exponent and $\delta > 0$ is a constant that depends on ε.

problem	approximation factor	running time
Feedback Arc Set	$1+\varepsilon$	$O^*((2-\delta)^n)$
Directed Cutwidth	2	$O^*(2^{\frac{\omega n}{3}})$
Optimal Linear Arrangement	$1.5+\varepsilon$	$O^*((2-\delta)^n)$
Directed Optimal Linear Arrangement	$2+\varepsilon$	$O^*((2-\delta)^n)$
Directed Pathwidth	2	$O^*(1.66^n)$

best known algorithm–even including constant-factor approximations. Moreover, the question whether e.g., Feedback Arc Set or Cutwidth can be solved faster than $O^*(2^n)$ have been explicit open problems for a long time [22, 24, 25, 12]. This reinforces our belief that it is time to study exponential-time approximation algorithms for vertex-ordering problems. The only example of such a study is in the case of (directed or undirected) bandwidth [13, 28]. Our results are summarized in Table 1. We believe that working on these vertex-ordering problems from a new angle will further enhance our understanding of them and we hope that there might be some interesting connections to find that use ideas from the vast knowledge acquired on these problems both from the world of polynomial-time approximations and from the world of exact exponential-time algorithms.

In the rest of this section, we survey a few different vertex-ordering problems and previous work related to them. Afterwards, we give an overview of our results and a high-level intuition of our main conceptual contribution.

Problem Definitions and Related Work

We now introduce the different problems used in this paper and briefly state relevant related work for each of them.

We start with a directed version of Minimum Cut with an additional size constraint.

Directed Minimum $(k, n-k)$-Cut

Input: A directed graph G and an integer k.

Task: Find a set L of exactly k vertices such that the number of arcs from $V \setminus L$ to L is minimized.

The undirected version of this problem was introduced by Bonnet et al. [7], who showed fixed-parameter tractability of the problem when parameterized by $k + \mathrm{OPT}$, where OPT is the minimum number of arcs from $V \setminus L$ to L for any set L of size k. We are not aware of any mentioning of the directed problem anywhere. We also study a weighted generalization where we are given an additional non-negative arc-weight function w and instead of minimizing the number of arcs from $V \setminus L$ to L, we instead minimize the weight of the arc set from $V \setminus L$ to L. Directed Minimum $(k, n-k)$-Cut will be at the heart of most of our algorithms.

We continue with Feedback Arc Set. It can be defined both in terms of a subset of arcs of an input graph and as a vertex-ordering problem. The former definition asks for a given directed graph $G = (V, A)$ to find a minimum-size set of arcs whose removal turns G into a directed acyclic graph (DAG). We give the definition as a vertex-ordering problem as this will turn out to be more useful for us later.

Feedback Arc Set
Input: A directed graph $G = (V, A)$.
Task: Find an ordering π of V minimizing the number of backward arcs with respect to π.

Feedback Arc Set can be solved in $O(2^n n^2)$ time by the Held-Karp algorithm [23]. The weighted version of this problem asks for an ordering such that the sum of weights of all backward arcs is minimized. The best known polynomial-time approximaiton algorithm (for both the weighted and unweighted variants) achieves an approximation factor of $O(\log(n) \log\log(n))$ [18] and any approximation better than 1.36 in polynomial time would refute $P \neq NP$ [29, 14]. As stated above, we achieve a $(1+\varepsilon)$-approximation for Feedback Arc Set faster than $O(2^n)$ for any $\varepsilon > 0$. We mention that some of the methods used for this result can also be applied to other vertex-ordering problems (to obtain constant-factor approximations). We list a few such problems in the following.

In Optimal Linear Arrangement (also known as Minimum Linear Arrangement), one is asked for an ordering of a graph that minimizes the sum of stretches of all (backward) arcs. Equivalently, the problem asks for an ordering minimizing the sum of cut sizes over all positions. In the following, we denote by cut^i_π for an ordering π of the vertices and an integer i, the set of all edges (or arcs) from a vertex after position i to a vertex at position at most i.

Optimal Linear Arrangement
Input: An directed or undirected graph G with vertex set V.
Task: Find an ordering π of V minimizing $\sum_{i=1}^{n-1} |\,\mathrm{cut}^i_\pi\,|$.

In the weighted setting, we want to minimize $\sum_{i=1}^{n-1} \sum_{e \in \mathrm{cut}^i_\pi} w(e)$. Optimal Linear Arrangement can be solved in $O^*(2^n)$ time [6] and the best known polynomial-time algorithm achieves an approximation factor in $O(\sqrt{\log(n)} \log\log(n))$ [20, 10]. It remains NP-hard even when restricted to interval graphs but there it can be 2-approximated in polynomial time [11].

In the problem Directed Cutwidth[2], we are looking for an ordering that minimizes the maximum size of any cut^i_π. Formally, it is defined as follows.

Directed Cutwidth
Input: A directed graph $G = (V, A)$.
Task: Find an ordering π of V minimizing $\max_{i \in [n-1]} |\,\mathrm{cut}^i_\pi\,|$.

In the weighted setting, we want to minimize the weight $\max_{i \in [n-1]} \sum_{e \in \mathrm{cut}^i_\pi} w(e)$ instead of the cardinality. Directed Cutwidth can be solved in $O(2^n n^2)$ time and the weighted version can be solved in $O(2^n n^2 \log(w_{\max}))$ time by the Held-Karp algorithm [23]. The best known approximation factor achieved by a polynomial-time algorithm is $O(\log^{1.5}(n))$ [3]. A quantum algorithm running in $O(1.817^n)$ time is also known [2] and if the input graph is undirected, then the problem admits a $\log^{1+o(1)}(n)$-approximation [5].

Finally, we study Directed Pathwidth. It is a directed generalization of Pathwidth which itself is variation of treewidth where the tree decomposition is restricted to being a path. It is known that Pathwidth has an equivalent definition in terms of vertex orderings

[2] We mention in passing that there is some inconsistency in the existing literature regarding the name Directed Cutwidth. An alternative definition takes a directed acyclic graph as input and searches for a topological ordering of the vertices minimizing the (undirected) cutwidth.

and this remains true also for the directed variant [30]. For a position i, consider the number of vertices v that appear before position i that have in-neighbors that appear after position i. The equivalent definition of Directed Pathwidth asks for an ordering that minimizes the maximum quantity over all positions. The formal definition in terms of vertex orderings is as follows.

Directed Pathwidth
Input: A directed graph $G = (V, A)$.
Task: Find an ordering π of V minimizing

$$\max_{i \in [n-1]} |\{v \in V \mid \pi(v) \leq i \wedge (\exists u \in V.\ \pi(u) > i \wedge (u, v) \in A)\}|.$$

The fastest exact algorithm for Directed Pathwidth runs in $O(1.89^n)$ time [30] and the best known approximation factor of a polynomial-time algorithm is $O(\log^{1.5}(n))$ [19]. The undirected pathwidth can be $\log^{1+o(1)}(n)$-approximated in polynomial time [5].

Overview

In this work, we initiate the study of exponential-time approximation algorithms for vertex-ordering problems. A recurring theme in our algorithms is a "balanced-cut" strategy, which results in a running time that is strictly faster than $O^*(2^n)$ and which we consider our main conceptual contribution. We briefly describe the main idea here. We first show how to find a partition of the set of vertices in a graph into two sets L and R of given sizes such that the number of arcs from R to L is (approximately) minimized, that is, we solve Directed Minimum $(k, n-k)$-Cut.[3] Depending on the specific problem at hand, we then either solve the two instances induced by these sets of vertices exactly or approximately. Finally, we show how to combine the two partial solutions with the computed cut between R and L. For Feedback Arc Set, we design an additional "self-improving" $(1+\varepsilon)$-approximation algorithm by combining the above approach with a novel bootstrapping strategy.

The rest of this work is structured as follows. In Section 2, we introduce notation and concepts used in this work. We show our results for unweighted problems in Section 3 and for weighted problems in Section 4. We conclude with Section 5.

2 Preliminaries

For a positive integer $n \in \mathbb{N}$, we denote by $[n]$ the set $\{1, 2, \ldots, n\}$. We use standard graph-theoretic notation. In particular, for a given graph $G = (V, E)$ and a set $V' \subseteq V$ of vertices, we use $G[V']$ to denote the subgraph of G induced by V'. To avoid confusion, we will use the term *edges* for undirected graphs and the term *arcs* for directed graphs. Moreover, we will refer to the edge set of an undirected input graph by E and the arc set of a directed input graph by A. All graphs in this work are simple, that is, they do not contain self-loops or parallel edges/arcs (but we allow in the directed case both the arc (u, v) and the arc (v, u) to be present at the same time for a pair of vertices $u \neq v$). We always denote the number of vertices in the input graph by n and the number of edges/arcs by m. We say that a graph is edge-weighted (respectively arc-weighted) if we are given an additional weight function $w \colon E \to \mathbb{N}$ ($w \colon A \to \mathbb{N}$). The weight of an edge $e \in E$ is then $w(e)$ and the weight of a set $E' \subseteq E$ of edges is the sum of edge weights of edges in E.

[3] We mention that we do not always just consider a single partition. Sometimes, we iterate over many such partitions and return the best solution over all considered partitions.

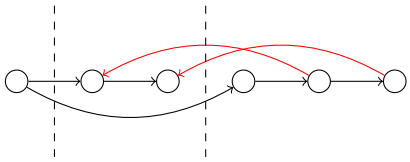


Figure 1 An example for cut_π in the directed setting. The vertices are ordered by some ordering π from left to right. The cuts cut_π^1 and cut_π^3 are depicted by dashed lines. The cut cut_π^1 is empty and the cut cut_π^3 consists of all the red arcs. The stretch of both red arcs is 3.

A *ordering* π of the vertices of a graph G with vertex set V is a bijection between V and $|V|$. For a directed graph $G = (V, A)$ and an ordering π of V, we say that an arc (u, v) is forward with respect to π if $\pi(u) < \pi(v)$ and the arc is backward with respect to π otherwise. The stretch of an edge $\{u, v\}$ (or an arc (u, v)) is $|\pi(u) - \pi(v)|$. For an integer $i \in [n]$, we say the cut at position i with respect to π is $\mathrm{cut}_\pi^i = \{\{u, v\} \in E \mid \pi(u) > i \wedge \pi(v) \leq i\}$. In the directed case, we use the definition $\mathrm{cut}_\pi^i = \{(u, v) \in A \mid \pi(u) > i \wedge \pi(v) \leq i\}$. See Figure 1 for an example.

We assume familiarity with the Bachmann–Landau notation (also known as big-O notation) and we use O^* to hide polynomial factors in the input size. We note, that this on the one hand hides polynomial factors in $\log(w_{\max})$, where $w_{\max}$ is the largest weight in the input graph. On the other hand, $O^*(\alpha^n) \subseteq O((\alpha + \varepsilon)^n)$ for any $\alpha \geq 1$ and any $\varepsilon > 0$.

3 Unweighted Problems

In this section, we present our algorithms for unweighted input graphs. We present all algorithms for the directed case but mention that it is straight-forward to adapt all of these algorithms to the undirected case. Most of our results are based on the following faster algorithm for Directed Minimum $(k, n-k)$-Cut which is a generalization of a similar algorithm for Maximum Cut by Williams [36]. Here and in the following, $\omega < 2.373$ is the matrix-multiplication exponent.

▶ **Proposition 1.** *Directed Minimum $(k, n-k)$-Cut can be solved in $O^*(\min(2^{\frac{\omega n}{3}}, n^{\omega k}))$ time.*

Proof. Let $(G = (V, A), k)$ be an input instance of Directed Minimum $(k, n-k)$-Cut. We partition V arbitrarily into three sets V_1, V_2, and V_3 of (roughly) equal size. Next, we guess in $O(n^2)$ time how many vertices of V_1, V_2, and V_3 belong to an optimal solution L. Let these numbers be k_1, k_2 and $k_3 = k - k_1 - k_2$. Next, we build a graph H as follows. For each $i \in [3]$, we introduce a vertex v_j^i for each subset $V_j^i \subseteq V_i$ with $|V_j^i| = k_i$. Let $\delta(v_j^i) = |\{(u, w) \in A \mid u \in V_i \setminus V_j^i \wedge w \in V_j^i\}|$ be the number of arcs from $V_i \setminus V_j^i$ to V_j^i. Note that H contains $O(\min(2^{n/3}, \binom{n/3}{k}))$ vertices. For each pair of vertices $v_j^i, v_{j'}^{i'}$ with $i \neq i'$, we add an edge between them. We set the weight of this edge to

$$|\{(u, w) \in A \mid u \in V_i \setminus V_j^i \wedge w \in V_{j'}^{i'}\}|$$
$$+ |\{(u, w) \in A \mid u \in V_{i'} \setminus V_{j'}^{i'} \wedge w \in V_j^i\}| + \frac{\delta(v_j^i) + \delta(v_{j'}^{i'})}{2}.$$

Note that each edge weight is at most n^2. Next, for each pair $i \neq i'$, we guess the weight of the edge $\{v_j^i, v_{j'}^{i'}\}$ in a minimum-weight triangle in H (without guessing the indices j and j'). Note that there are n^6 possible guesses. For each guess, we compute in time linear in the size of H

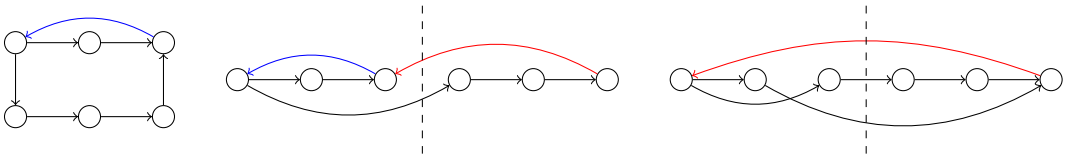


Figure 2 An example of Feedback Arc Set. The input graph is given on the left and the middle/right show two optimal solutions for Directed Minimum $(k, n-k)$-Cut with $k = 3$ (red arcs). The optimal solution to Feedback Arc Set is shown in blue (and coincides with the red arc on the right). If we use the cut in the middle picture, then we can only find a 2-approximation while the cut on the right leads to an optimal solution.

a subgraph which only contains the edges corresponding to the current guess. We then check whether this subgraph contains a triangle in $O^*((\min(2^{\frac{n}{3}}, \binom{n/3}{k}))^\omega) = O^*(\min(2^{\frac{\omega n}{3}}, n^{\omega k}))$ time [27]. The algorithm also finds a triangle if one exists. After going through all possible guesses, we report the triangle of minimum weight. Note that this weight corresponds by construction to a minimum directed $(k, n-k)$-cut such that $|L \cap V_i| = k_i$ for all $i \in [3]$. Since we iterate over all possible choices of k_i, we always find a minimum-weight directed $(k, n-k)$-cut. The total running time is in $O^*(\min(2^{\frac{\omega n}{3}}, n^{\omega k}))$. ◀

3.1 Feedback Arc Set

We next show how to use Proposition 1 to design a $(1+\varepsilon)$-approximation for Feedback Arc Set for any $\varepsilon > 0$ that is faster than the $O^*(2^n)$-time Held-Karp algorithm. We first show how to find a 2-approximation in $O^*(2^{\frac{\omega n}{3}})$ time and then show how to use a $(1+\frac{1}{k})$-approximation to find a $(1+\frac{1}{k+1})$-approximation.

▶ **Lemma 2.** *Feedback Arc Set can be 2-approximated in $O^*(2^{\frac{\omega n}{3}})$ time.*

Proof. Let $G = (V, A)$ be an input graph for Feedback Arc Set. We first compute an optimal solution $L \subseteq V$ of the instance $(G, \lfloor \frac{n}{2} \rfloor)$ of Directed Minimum $(k, n-k)$-Cut in $O^*(2^{\frac{\omega n}{3}})$ time using Proposition 1. Observe that $|L| = \lfloor \frac{n}{2} \rfloor$, and that the number of arcs from $V \setminus L$ to L is a lower bound for the size of a minimum feedback arc set in G. Let this set of arcs be A' and let the set of arcs from L to $V \setminus L$ be A''. Next, we solve the two instances $G[L]$ and $G[V \setminus L]$ of Feedback Arc Set optimally in $O^*(2^{\frac{n}{2}})$ time [23]. See Figure 2 for an example. Since a feedback arc set in a graph is also a feedback arc set in any subgraph, it holds that the optimal solutions in $G[L]$ and $G[V \setminus L]$ are combined a lower bound for the size of a feedback arc set in G. Hence, placing the vertices of L in the computed order before the vertices of $V \setminus L$ in the computed order yields a 2-approximation as each arc in A' contributes one and the arcs in A'' do not contribute anything. The running time is in $O^*(2^{\frac{\omega n}{3}} + 2^{\frac{n}{2}}) = O^*(2^{\frac{\omega n}{3}})$. ◀

We next show how to use any constant-factor approximation for Feedback Arc Set that is faster than $O^*(2^n)$ to compute a better constant factor approximation that is also faster than $O^*(2^n)$. This "boosting technique" results in the following approximation scheme.

▶ **Theorem 3.** *For each constant $\varepsilon > 0$, there is a $\delta_\varepsilon > 0$ such that Feedback Arc Set can be $(1+\varepsilon)$-approximated in $O((2-\delta_\varepsilon)^n)$ time.*

Proof. We prove this statement by showing via induction that for each $k \geq 1$, there is a $\delta_k > 0$ such that Feedback Arc Set can be $(1+\frac{1}{k})$-approximated in $O((2-\delta_k)^n)$ time. Since any constant $\varepsilon > 0$ is larger than $\frac{1}{k}$ for some constant value of k, this will prove the

theorem. Note that the base case $k = 1$ is proven by Lemma 2. It remains to show the induction step, that is, assuming that the statement is true for some value of k, we need to show that the statement is also true for $k + 1$. To this end, let k be arbitrary but fixed and assume that Feedback Arc Set can be $(1 + \frac{1}{k})$-approximated in $(2 - \delta_k)^n$ time for some $\delta_k > 0$. Let γ be the solution to the equation

$$\frac{\gamma \log(\gamma) - (\gamma - 1)\log(\gamma - 1)}{\gamma - 1} = 1 - \log(2 - \delta_k),$$

where all logarithms use base 2, and let $\alpha = \frac{1}{\gamma}$.[4] Note that $\gamma \geq 2$ exists for each $0 < \delta_k < 1$ as the right side of the equations is some value between zero and one and the left side evaluates to 2 for $\gamma = 2$ and tends towards 0 for increasing values of γ as shown next. Note that the left side of the previous equation is equivalent to $(\log(\gamma) - \log(\gamma - 1)) + \frac{\log(\gamma)}{\gamma - 1}$. The first summand describes the derivative of the logarithm (which is $\frac{1}{\ln(2)\gamma}$) and therefore tends towards 0 for increasing values of γ. The second summand also clearly tends towards zero as the logarithm grows sublinearly.

We next describe the $(1 + \frac{1}{k+1})$-approximation for Feedback Arc Set. Let $G = (V, A)$ be the input graph and let n be the number of vertices in V. For the sake of simplicity, we will assume that αn is an integer to avoid dealing with rounding issues (which will vanish in the Bachmann-Landau notation anyway). Bodlaender et al. [6] showed how to solve Feedback Arc Set exactly in $O^*(2^n)$ time. Importantly, their algorithm iterates over all induced subgraphs (in order of increasing size), and computes the optimal solution for each induced subgraph in polynomial time for each subgraph. We use their algorithm for all induced subgraphs with at most αn vertices. Let $\mathcal{V}$ be the set of all sets of vertices of size exactly αn. Using the algorithm by Bodlaender et al. [6] to compute the optimal solution for all graphs induced by sets in $\mathcal{V}$ takes $O^*(\binom{n}{\alpha n})$ time. In the same time, we can also compute for each $V' \in \mathcal{V}$ the number of arcs in $\{(u, v) \mid u \in V \setminus V' \wedge v \in V'\}$. Let $a(V')$ denote this number and let $V^* \in \mathcal{V}$ be a set minimizing $a(V^*)$. We then use the algorithm by Bodlaender et al. [6] to solve the instance $G[V \setminus V^*]$ optimally in $O^*(2^{(1-\alpha)n})$ time and for all other $V' \in \mathcal{V}$, we use the $(1 + \frac{1}{k})$-approximation for $G[V \setminus V']$ that runs in $O((2 - \delta_k)^{(1-\alpha)n})$ time. Finally for each $V' \in \mathcal{V}$, we compute the sum of the optimal value for $G[V']$ plus $a(V')$ plus the size of the computed solution for $G[V \setminus V']$. We return the smallest number found (or the set associated to the smallest number if we want to return the feedback arc set).

It remains to show that our algorithm always produces a $(1 + \frac{1}{k+1})$-approximation and to analyze the running time. For the approximation factor, let F be a feedback arc set of G of minimum size and let $\mathrm{OPT} = |F|$. We distinguish between the following two cases. Either $a(V^*) \leq \frac{1}{k+1}\,\mathrm{OPT}$ or $a(V^*) > \frac{1}{k+1}\,\mathrm{OPT}$. In the first case, note that the solution we computed for V^* consists of optimal solutions for $G[V^*]$ and $G[V \setminus V^*]$ plus $a(V^*)$. Since the first two summands are combined at most OPT and $a(V^*) \leq \frac{1}{k+1}\,\mathrm{OPT}$, we indeed return a feedback arc set of size at most $(1 + \frac{1}{k+1})\,\mathrm{OPT}$. If $a(V^*) > \frac{1}{k+1}\,\mathrm{OPT}$, then note that for each $V' \in \mathcal{V}$ it holds that $a(V') > \frac{1}{k+1}\,\mathrm{OPT}$. Moreover, since the graph $G' = (V, A \setminus F)$ is acyclic, it has a topological ordering. Let V'' be the set of the first αn vertices in that topological ordering. Let d be the size of a minimum feedback arc set in $G[V \setminus V'']$ and let $c = \frac{d}{\mathrm{OPT}}$. Note that $c \leq \frac{k}{k+1}$. Finally, the solution we compute for V'' has size

[4] We note that γ is roughly $-\frac{W(-\frac{(2-\delta_k)(1-\log(2-\delta_k))}{2})}{1-\log(2-\delta_k)}$, where W is the Lambert W function (also known as the product logarithm). It is known that this function cannot be expressed in terms of elementary functions.

$$(1-c)\,\mathrm{OPT}+(1+\frac{1}{k})c\,\mathrm{OPT} \leq \mathrm{OPT}-c\,\mathrm{OPT}+c\,\mathrm{OPT}+\frac{1}{k}\frac{k}{k+1}\,\mathrm{OPT} = (1+\frac{1}{k+1})\,\mathrm{OPT}\,.$$

Thus, we also return a $(1+\frac{1}{k+1})$-approximation in this case.

Let us now analyze the running time. Recall that $\gamma = \frac{1}{\alpha}$. Computing $\mathcal{V}$, $a(V')$, and the optimal solution for $G[V']$ for each $V' \in \mathcal{V}$ take $O^*(\binom{n}{\alpha n})$ time. Solving the instance $G[V \setminus V^*]$ optimally takes $O^*(2^{(1-\alpha)n})$ time. Computing the approximate solutions for all sets in $\mathcal{V}$ takes overall $O^*(\binom{n}{\alpha n} \cdot (2-\delta_k)^{(1-\alpha)n})$ time. By standard arguments, $\binom{n}{\alpha n} \in O((\frac{\gamma^\gamma}{(\gamma-1)^{\gamma-1}})^{\alpha n})$ [35]. Hence

$$\begin{aligned}
O(\binom{n}{\alpha n} \cdot (2-\delta_k)^{(1-\alpha)n}) &= O(2^{\log(\frac{\gamma^\gamma}{(\gamma-1)^{\gamma-1}})^{\alpha n})} \cdot 2^{\log((2-\delta_k)^{(1-\alpha)n})}) \\
&\subseteq O(2^{\alpha n(\log(\gamma^\gamma)-\log((\gamma-1)^{\gamma-1}))+(1-\alpha)n\log(2-\delta_k)}) \\
&\subseteq O(2^{(1-\alpha)n\left(\frac{\alpha}{1-\alpha}(\gamma\log(\gamma)-(\gamma-1)\log(\gamma-1))+\log(2-\delta_k)\right)}) \\
&\subseteq O(2^{(1-\alpha)n\left(\frac{\gamma\log(\gamma)-(\gamma-1)\log(\gamma-1)}{\gamma-1}+\log(2-\delta_k)\right)}) \\
&\subseteq O(2^{(1-\alpha)n}),
\end{aligned}$$

where again all logarithms use base 2. The last equality is due to the definition of γ. Thus, the total running time of our algorithm is $O^*(2^{(1-\alpha)n})$. This is in $O((2-\delta_{k+1})^n)$ time for any $0 < \delta_{k+1} < 2-2^{1-\alpha}$. This completes the induction step and concludes the proof. ◀

3.2 Cutwidth

We continue with Directed Cutwidth. We again show how to use Proposition 1 to compute a 2-approximaiton.

▶ **Proposition 4.** *Directed Cutwidth can be 2-approximated in $O^*(2^{\frac{\omega n}{3}})$ time.*

Proof. Let $G = (V, A)$ be an input to Directed Cutwidth. We first find an optimal solution L to the instance $(G, \lfloor \frac{n}{2} \rfloor)$ of Directed Minimum $(k, n-k)$-Cut in $O^*(2^{\frac{\omega n}{3}})$ time using Proposition 1. Let $R = V \setminus L$ and let A^* be the set of arcs (u, v) with $u \in R$ and $v \in L$. Note that the size of A^* is by definition a lower bound for the directed cutwidth of G. Next, we can solve $G[L]$ and $G[R]$ optimally in $O^*(2^{\frac{n}{2}})$ time [23]. Since adding a vertex and/or arc to a graph cannot decrease its directed cutwidth, we have that the cutwidth of $G[L]$ and $G[R]$ are both at most the cutwidth of G. Hence, placing all vertices in L according to the computed solution for $G[L]$ and then all vertices in R according to the solution for $G[R]$ gives an ordering for the graph $G' = (V, A \setminus A^*)$ that shows that the cutwidth of G' is at most the cutwidth of G. See Figure 3 for an example. Adding the set A^* to G' can increase the cutwidth by at most $|A^*|$ and adding arcs (u, v) with $u \in L$ and $v \in R$ does not increase the directed cutwidth. Thus, the computed ordering is a 2-approximation as the size of A^* is at most the cutwidth of G as shown above. Note that the running time is in $O^*(2^{\frac{\omega n}{3}} + 2^{\frac{n}{2}}) = O^*(2^{\frac{\omega n}{3}})$. This concludes the proof. ◀

Unfortunately, due to the inherent maxima nature of Directed Cutwidth, we do not know how to apply a boosting technique similar to Theorem 3 to get an approximation factor smaller than 2.

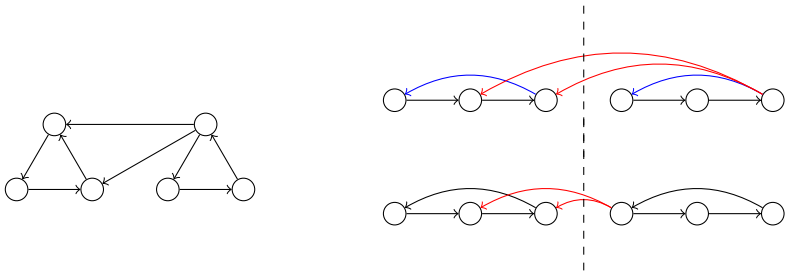


Figure 3 An example of Directed Cutwidth. The input instance is depicted on the left with the exception that all vertices in the left triangle have arcs towards all vertices in the right triangle unless the respective reverse arc is present. We decided to not show these arcs for the sake of readibility. On the right, two optimal solutions to Directed Minimum $(k, n-k)$-Cut with $k = 3$ are shown (red arcs) with optimal orderings for the two respective subproblems of Directed Cutwidth are depicted. The ordering shown on the top has a directed cutwidth of 3 (e.g. at positions 4 and 5) and the ordering on the bottom has a directed cutwidth of 2.

3.3 Optimal Linear Arrangement

In this section, we study Optimal Linear Arrangement. We show how to compute a $(2+\varepsilon)$-approximation for Directed Optimal Linear Arrangement for each $\varepsilon > 0$ in $O((2-\delta_\varepsilon)^n)$ time for some $\delta_\varepsilon > 0$ only depending on ε. We then show how to improve the algorithm in the undirected setting to achieve a $(\frac{3}{2}+\varepsilon)$-approximation in the same time.

▶ **Proposition 5.** *For every $0 < \alpha < 1$, there is a $(1 + \frac{1}{(1-\alpha)})$-approximation for Directed Optimal Linear Arrangement that runs in $O^*(2^{\frac{\omega n}{3}} + 2^{(1-\frac{\alpha}{2})n})$ time, where ω is the matrix-multiplication exponent.*

Proof. Let $G = (V, A)$ be an input to Directed Optimal Linear Arrangement. We use Proposition 1 to solve Directed Minimum $(k, n-k)$-Cut for each $\frac{\alpha n}{2} \leq k \leq n - \frac{\alpha n}{2}$. This takes $O^*(2^{\frac{\omega n}{3}})$ time. Let k' be a value where the number of arcs from $V \setminus L$ to L are minimum for some set L of size k'. Let L' be a set of vertices of size k' minimizing the number of arcs from $R' = V \setminus L'$ to L' and let A' be the set of arcs from R' to L'. By the pigeonhole principle, the size of A' is at most $\frac{\mathrm{OPT}}{(1-\alpha)n}$ where OPT is the optimal directed-optimal-linear-arrangement value for G (the value we want to minimize in Directed Optimal Linear Arrangement). Note that $|L'|, |R'| \leq (1 - \frac{\alpha}{2})n$. Next, we solve the instances $G[L]$ and $G[R]$ optimally in $O^*(2^{(1-\frac{\alpha}{2})n})$ time using the algorithm by Bodlaender et al. [6]. To get an approximation for the whole graph G, we first order all vertices in L according to the computed solution for $G[L]$ and then all vertices in R according to the computed solution for R. Note that this is an optimal solution for the graph $G' = (V, A \setminus A')$. It remains to add the arcs between L and R. Note that each edge in A' can contribute at most n and therefore the edges in A' can contribute at most $\frac{\mathrm{OPT}}{1-\alpha}$ in total. Arcs from L to R do not contribute as they are forward arcs in the computed ordering. Added to the cost for G' (which is at most OPT), this yields the claimed approximation factor. The running time is in $O^*(2^{\frac{\omega n}{3}} + 2^{(1-\frac{\alpha}{2})n})$. ◀

We mention that the above result yields a 2.72-approximation in $O^*(2^{\frac{\omega n}{3}})$ time, assuming the currently known value of $\omega < 2.372$. We next show how to improve the approximation factor to $(\frac{3}{2}+\varepsilon)$ in the undirected setting. The general strategy is the same but in the end, we use the fact that an ordering and its reverse have the same value in terms of Optimal Linear Arrangement and leverage this fact to argue that contribution of each edge between the two parts is at most $\frac{n}{2}$ on average.

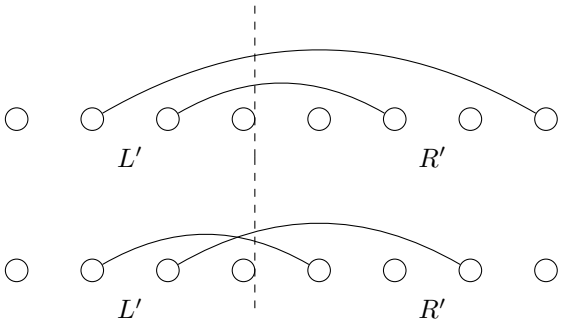


Figure 4 An example of Optimal Linear Arrangement where the vertices are divided into two parts L' and R' with two edges between the two parts and some ordering within the two parts is fixed. For the sake of simplicity, we do not show edges within the two parts. The two orderings only differ in that we reverse the ordering of the vertices in R'. The average length within L' of the two edges between L' and R' is 1.5 in both orderings. The average length within R' is 3 in the above ordering and $2'$ in the below ordering.

▶ **Proposition 6.** *For each constant $0 < \alpha < 1$, there is a $(1 + \frac{1}{2(1-\alpha)})$-approximation for Optimal Linear Arrangement that runs in $O^*(2^{\frac{\omega n}{3}} + 2^{(1-\frac{\alpha}{2})n})$ time, where ω is the matrix-multiplication exponent.*

Proof. Let $G = (V, E)$ be an input graph for Optimal Linear Arrangement. We start the same as in Proposition 5, that is, we solve Minimum $(k, n-k)$-Cut for each $\frac{\alpha n}{2} \leq k \leq \frac{n}{2}$, define sets L', R', and E' and value OPT, and solve $G[L']$ and $G[R']$ optimally in $O^*(2^{\frac{\omega n}{3}} + 2^{(1-\frac{\alpha}{2})n})$ time overall. Let $i = |L'|$. We next test for both L' and R' whether the average length of edges in E' is smaller if we order the vertices according to the computed solution or in the reverse order. To this end, we pretend that the endpoint outside the respective part is fixed at position i. See Figure 4 for an example. We pick for each of the two the ordering which results in shorter average lengths and observe that the average length of an edge in E' is now at most $\frac{n}{2}$ as the average length "in L'" is at most $\frac{|L'|-1}{2}$, the average length "in R'" is at most $\frac{|R'|+1}{2}$, and $|L'| + |R'| = n$. The combined cost for all edges in E' is therefore at most $\frac{\text{OPT}}{(1-\alpha)n} \cdot \frac{n}{2} = \frac{\text{OPT}}{2(1-\alpha)}$ and added to the cost for G' (which is at most OPT), this yields the claimed approximation factor. The running time is in $O^*(2^{\frac{\omega n}{3}} + 2^{(1-\frac{\alpha}{2})n})$. ◀

The above results yields a 1.86-approximation in $O^*(2^{\frac{\omega n}{3}})$ time using the currently known value of $\omega \approx 2.372$.

3.4 Pathwidth

We conclude this section with a 2-approximation for Directed Pathwidth in $O(1.66^n)$ time. We point out that this result stands out from the previous in two ways: Directed Pathwidth can be solved in $O(1.89^n)$ time and our algorithm is not based on Proposition 1.

▶ **Theorem 7.** *Directed Pathwidth can be 2-approximated in $O(1.66^n)$ time.*

Proof. Let $G = (V, A)$ be the input graph for which we want to approximate the pathwidth, let $\alpha \approx 0.204$ be the solution to the equation $1.89^{1-\alpha} = (\frac{1}{\alpha})^{\alpha}(\frac{1}{1-\alpha})^{1-\alpha}$, and let OPT be the (unknown) directed pathwidth of G. We first use the algorithm by Bodlaender et al. [6] to find a set V' of αn vertices (and an ordering of these vertices) such that independent of the ordering of the remaining $(1-\alpha)n$ vertices, it holds for each position i that at most OPT

vertices before position $\min(i, \alpha n)$ have in-neighbors after position i. This can be done in time $O^*((\frac{1}{\alpha})^{\alpha n}(\frac{1}{1-\alpha})^{(1-\alpha)n}) = O^*(1.89^{(1-\alpha)n})$. Next, we solve the instance $G[V \setminus V']$ of Directed Pathwidth optimally in $O^*(1.89^{(1-\alpha)n})$ time [30]. We return the ordering that places the vertices in V' first (in the order that was computed before) and then all vertices in $V \setminus V'$ in the order corresponding to the solution for $G[V \setminus V']$. Note that it holds for each position $i > \alpha n$ that at most OPT vertices before position αn have in-neighbors after position i and at most OPT vertices between position $\alpha n + 1$ and i have in-neighbors after position i, that is, the computed ordering is a 2-approximation. The running time is in $O^*(1.89^{(1-\alpha)n}) \approx O^*(1.89^{(0.796)n}) \subseteq O(1.66^n)$. This concludes the proof. ◀

4 Weighted Problems

In this section, we consider weighted generalizations of the problems studied in the last section. We denote the largest weight in the input by $w_{\max}$. We mention that if $w_{\max}$ is polynomially upper-bounded in n, then all of our previous results generalize in a straight-forward way. Hence, we mainly focus on cases where $w_{\max}$ is large compared to n. An assumption that is often made when dealing with weighted problems is that all numbers can be added and subtracted in constant time. However, in the realm of very large weights, this assumption is highly unrealistic as even reading the weight from input takes $O(\log(w_{\max}))$ time (which we sometimes hide in the O^*-notation). We decided to not make the above assumption which basically adds an additional $O(\log(w_{\max}))$ factor to all running times.

We mention that we do not study a weighted version of Directed Pathwidth. To the best of our knowledge, this problem has only been studied once [32] and our algorithm behind Theorem 7 computes a 2-approximation even in the weighted setting if we have an exact (slower) algorithm. We suspect that the $O(1.89^n)$-time algorithm by Kitsunai et al. [30] generalizes to the weighted setting but this was never proven and proving it here would go beyond the scope of this paper.

We start with an arc-weighted generalization of Directed Minimum $(k, n-k)$-Cut. We mention that if this generalization can be solved in $O^*(2^{\frac{\omega n}{3}})$ time, then our previous algorithms for Feedback Arc Set, Directed Optimal Linear Arrangement, and Directed Cutwidth can all easily be generalized to the weighted setting. Unfortunately, even for (undirected) Weighted Max Cut, no such algorithm is known. However, Williams [36] showed how to compute a $(1+\varepsilon)$-approximation for Weighted Max Cut in $O^*(2^{\frac{\omega n}{3}}(\frac{1}{\varepsilon})^3)$ time. We will generalize this result to Weighted Directed Minimum $(k, n-k)$-Cut and show how to use this approximation for the other problems.

▶ **Proposition 8.** *For any $\varepsilon > 0$, Weighted Directed Minimum $(k, n-k)$-Cut can be $(1+\varepsilon)$-approximated in $O(2^{\frac{\omega n}{3}} \cdot \log(w_{\max}) \cdot (\frac{1}{\varepsilon})^2 \cdot k^2)$ time.*

Proof. Let $(G = (V, A), k)$ be an input instance and let $\varepsilon > 0$ be an arbitrary but fixed value. We partition V arbitrarily into three sets V_1, V_2, and V_3 of (roughly) equal size. Next, we guess in $O(k^2)$ time how many vertices of V_1, V_2, and V_3 belong to an optimal solution A. Let these numbers be k_1, k_2 and $k_3 = k - k_1 - k_2$. Next, we build an undirected graph H with edge weights as follows. For each $i \in [3]$, we introduce a vertex v_j^i for each subset $V_j^i \subseteq V_i$ with $|V_j^i| = k_i$. Note that the number of vertices in H is in $O(2^{n/3})$. For each vertex v_j^i, let $d(v_j^i)$ denote the sum of weights of all arcs from vertices in V_j^i to vertices in $V_i \setminus V_j^i$ in G. For each pair of vertices $v_j^i, v_{j'}^{i'}$ with $i \neq i'$, we add an edge between them of weight

$$\frac{d(v_j^i) + d(v_{j'}^{i'})}{2} + \sum_{\{(u,w) \in A \mid u \in V_j^i \wedge w \in V_{i'} \setminus V_{j'}^{i'}\} \cup \{(u,w) \in A \mid u \in V_{j'}^{i'} \wedge w \in V_i \setminus V_j^i\}} w((u,w)).$$

Using an algorithm by Zwick [38] (see also Williams [36]), we find a $(1+\varepsilon)$-approximation for the minimum-weight triangle in H in $O(2^{\frac{\omega n}{3}} \frac{1}{\varepsilon} \log(\frac{n^2 w_{\max}}{\varepsilon})) \subseteq O(2^{\frac{\omega n}{3}} (\frac{1}{\varepsilon})^2 \log(w_{\max}))$ time. Note that $\log(n^2 w_{\max}) \leq \log({w_{\max}}^3) \leq 3\log(w_{\max})$ as we assume $w_{\max} > n$. Moreover, each triangle in H corresponds to a directed $(k, n-k)$-cut in H of the same weight and each directed $(k, n-k)$-cut with k_i vertices in V_i for each $i \in [3]$ corresponds to a triangle of the same weight in H. Hence, we compute a $(1+\varepsilon)$-approximation for Weighted Directed Minimum $(k, n-k)$-Cut. The total running time is in $O(2^{\frac{\omega n}{3}} \log(w_{\max})(\frac{1}{\varepsilon})^2 k^2)$ and this concludes the proof. ◀

4.1 Feedback Arc Set

We next show how to use Proposition 8 to get faster approximation algorithms for the weighted variants of the other vertex-ordering problems we consider. We start with Weighted Feedback Arc Set. Similar to the unweighted case, we first show a constant factor approximation and then show a boosting technique to achieve any $(1+\varepsilon)$-approximation.

▶ **Proposition 9.** *Weighted Feedback Arc Set admits a 3-approximation that can be computed in $O(2^{\frac{\omega n}{3}} \log(w_{\max})k^2)$ time.*

Proof. Let $G = (V, A)$ be an input graph for Weighted Feedback Arc Set. We first compute a 2-approximation L for the instance $(G, \lfloor \frac{n}{2} \rfloor)$ of Weighted Directed Minimum $(k, n-k)$-Cut in $O(2^{\frac{\omega n}{3}} \log(w_{\max})k^2)$ time using Proposition 8 with $\varepsilon = 1$. Note that half the sum of weights of arcs from $R = V \setminus L$ to L is a lower bound for the size of a minimum feedback arc set. We refer to Figure 2 for an illustration and mention that this time the weight of all red arcs is a 2-approximation for an optimal cut. Next, we solve the two instances $G[L]$ and $G[R]$ of Weighted Feedback Arc Set optimally in $O(2^{\frac{n}{2}} n^2 \log(w_{\max}))$ time [23]. Placing the vertices of L in the computed order before the vertices of R in the computed order yields a 3-approximation as the weights of the feedback arcs within both instances are combined also a lower bound for the weight of a minimum-weight feedback arc set in G. The running time is in $O(2^{\frac{\omega n}{3}} \log(w_{\max})k^2)$ as $\frac{\omega}{3} > \frac{1}{2}$. ◀

We next present the boosting technique in the weighted setting. It is very similar to the unweighted setting with two minor adjustments: the formulas change slightly due to the fact that we start with a 3-approximation rather than a 2-approximation and an additional factor of $O(\log(w_{\max}))$ is used when using the algorithm by Bodlaender et al. [6]. We present the entire proof for the sake of completeness.

▶ **Theorem 10.** *For each $\varepsilon > 0$, there is a $\delta > 0$ such that Weighted Feedback Arc Set can be $(1+\varepsilon)$-approximated in $O((2-\delta)^n \cdot \log(w_{\max}))$ time.*

Proof. We prove this statement by showing via induction that for each $k \geq 1$, there is a $\delta > 0$ such that Weighted Feedback Arc Set can be $(1 + \frac{2}{k})$-approximated in $O((2-\delta)^n \log(w_{\max}))$ time. Since any $\varepsilon > 0$ is larger than $\frac{2}{k}$ for some value of k, this will prove the theorem. Note that the base case $k = 1$ is proven by Proposition 9. It remains to show the induction step, that is, assuming that the statement is true for some value of k, we need to show that the statement is also true for $k+1$. To this end, let k be arbitrary but fixed and assume that Weighted Feedback Arc Set can be $(1 + \frac{2}{k})$-approximated in $(2-\delta_k)^n$ time for some $\delta_k > 0$. Let γ be the solution to the equation

$$\frac{\gamma \log(\gamma) - (\gamma - 1)\log(\gamma - 1)}{\gamma - 1} = 1 - \log(2 - \delta_k),$$

where all logarithms use base 2, and let $\alpha = \frac{1}{\gamma}$.

We next describe how to get the $(1+\frac{2}{k+1})$-approximation for Weighted Feedback Arc Set. Let $G=(V,A)$ be the input graph and let n be the number of vertices in V. For the sake of simplicity, we will again assume that αn is an integer. Let $\mathcal{V}$ be the set of all vertices with exactly αn vertices. Using the algorithm by Bodlaender et al. [6], we compute the optimal solution for all graphs induced by sets in $\mathcal{V}$. This takes $O(\binom{n}{\alpha n}\operatorname{poly}(n)\log(w_{\max}))$ time. In the same time, we can also compute for each $V'\in\mathcal{V}$ the sum of weights of arcs in $\{(u,v)\mid u\in V\setminus V'\wedge v\in V'\}$. Let $a(V')$ denote this number and let $V^*\in\mathcal{V}$ be a set minimizing $a(V^*)$. We then use the algorithm by Bodlaender et al. [6] to solve the instance $G[V\setminus V^*]$ optimally in $O(2^{(1-\alpha)n}\operatorname{poly}(n)\log(w_{\max}))$ time and for all other $V'\in\mathcal{V}$, we use the $(1+\frac{2}{k})$-approximation for $G[V\setminus V']$ that runs in $O((2-\delta_k)^{(1-\alpha)n})$ time. Finally, we for each $V'\in\mathcal{V}$, we compute the sum of the optimal value for $G[V']$ plus $a(V')$ plus the size of the computed solution for $G[V\setminus V']$. We return the smallest number found.

It remains to show that our algorithm always produces a $(1+\frac{2}{k+1})$-approximation and to analyze the running time. For the approximation factor, let F be a feedback arc set of G of minimum weight OPT. We distinguish between the following two cases. Either $a(V^*)\leq\frac{2}{k+1}\operatorname{OPT}$ or $a(V^*)>\frac{2}{k+1}\operatorname{OPT}$. In the first case, note that the solution we computed for V^* consists of optimal solutions for $G[V^*]$ and $G[V\setminus V^*]$ plus $a(V^*)$. Since the first two summands are combined at most OPT and $a(V^*)\leq\frac{2}{k+1}\operatorname{OPT}$, we indeed return a feedback arc set of size at most $(1+\frac{2}{k+1})\operatorname{OPT}$. If $a(V^*)>\frac{2}{k+1}\operatorname{OPT}$, then note that for each $V'\in\mathcal{V}$ it holds that $a(V')>\frac{2}{k+1}\operatorname{OPT}$. Moreover, since the graph $G'=(V,A\setminus F)$ is acyclic, it has a topological ordering. Let V'' be the set of the first αn vertices in that topological ordering. Let d be the weight of a feedback arc set of minimum weight in $G[V\setminus V'']$. Note that $d\leq\frac{k-1}{k+1}\operatorname{OPT}$. The solution we compute for V'' has size $(\operatorname{OPT}-d)+(1+\frac{2}{k})d\leq(1+\frac{2(k-1)}{k(k+1)})\operatorname{OPT}<(1+\frac{2}{k+1})\operatorname{OPT}$. Thus also in this case, we return a $(1+\frac{2}{k+1})$-approximation.

We conclude with analyzing the running time. Recall that $\gamma=\frac{1}{\alpha}$. Computing $\mathcal{V}$, $a(V')$, and the optimal solution for $G[V']$ for each $V'\in\mathcal{V}$ takes $O(\binom{n}{\alpha n}\operatorname{poly}(n)\log(w_{\max}))$ time. Solving the instance $G[V\setminus V^*]$ optimally takes $O(2^{(1-\alpha)n}\operatorname{poly}(n)\log(w_{\max}))$ time. Approximating the solutions for all sets in $\mathcal{V}$ takes $O(\binom{n}{\alpha n}\cdot(2-\delta_k)^{(1-\alpha)n}\cdot\log(w_{\max}))$ time in total. As shown in the proof for Theorem 3, it holds that $O(\binom{n}{\alpha n}\cdot(2-\delta_k)^{(1-\alpha)n})\subseteq O(2^{(1-\alpha)n})$. Thus, the total running time of our algorithm is in $O^*(2^{(1-\alpha)n})$ and in $O((2-\delta)^n\log(w_{\max}))$ for any $0<\delta<2-2^{1-\alpha}$. This concludes the proof. ◀

4.2 Cutwidth and Optimal Linear Arrangement

In this final part, we generalize our algorithms for Directed Cutwidth and Directed Optimal Linear Arrangement to the weighted setting. Since all generalizations follow the same simple structure, we only show the result for Directed Cutwidth.

▶ **Proposition 11.** *Weighted Directed Cutwidth admits a $(2+\varepsilon)$-approximation in time $O(2^{\frac{\omega n}{3}}\log(w_{\max})(\frac{1}{\varepsilon})^2k^2)$.*

Proof. The proof is similar to the proof of Proposition 4. Let $(G=(V,A),w)$ be the input instance to Weighted Directed Cutwidth. We find a $(1+\varepsilon)$-approximation for the instance $(G,w,\lfloor\frac{n}{2}\rfloor)$ of Weighted Directed Minimum $(k,n-k)$-Cut using Proposition 8. This takes $O(2^{\frac{\omega n}{3}}\log(w_{\max})(\frac{1}{\varepsilon})^2k^2)$ time. Let L be the set of vertices returned by Proposition 1, let $R=V\setminus L$, and let $A'\subseteq A$ be the set of arcs from R to L. Note that $\frac{w(A')}{1+\varepsilon}$ is by definition a lower bound for the weighted directed cutwidth of G. Next, we can solve $G[L]$ and $G[R]$ optimally in $O^*(2^{\frac{n}{2}}\log(w_{\max}))$ time [23]. Since adding a vertex

and/or arc to a graph cannot decrease its cutwidth, we have that the weighted directed cutwidth of $G[L]$ and $G[R]$ are both at most the weighted directed cutwidth of G. Adding the edges from L to R also does not increase the cutwidth. Hence, placing all vertices in L according to the computing solution for $G[L]$ and then all vertices in R according to the solution for $G[R]$ gives an ordering for the graph $G' = (V, A \setminus A')$ that shows that the weighted directed cutwidth of G' is at most the weighted directed cutwidth of G. Adding the set A' to G' can increase the cutwidth by at most $w(A')$ and hence the computed ordering is a $(2+\varepsilon)$-approximation. The running time is in $O(2^{\frac{\omega n}{3}} \log(w_{\max})(\frac{1}{\varepsilon})^2 k^2)$ as $\frac{\omega}{3} > \frac{1}{2}$ and this concludes the proof. ◀

The generalizations for Optimal Linear Arrangement and Directed Optimal Linear Arrangement are basically the same and are hence omitted. We just mention that the slightly weaker running time $(1 - \frac{\alpha}{4})$ instead of $(1 - \frac{\alpha}{2})$ in the exponent comes from the fact that we only have a $(1+\varepsilon)$-approximation for Weighted Directed Minimum $(k, n-k)$-Cut (where we set $\varepsilon = \frac{\alpha}{2}$ and then compute the approximation for all $\frac{\alpha}{4}n \leq k \leq (1 - \frac{\alpha}{4})n$ in order to compensate. The correctness then follows from the fact that $\frac{1+\frac{\alpha}{2}}{1-\frac{\alpha}{2}} \leq \frac{1}{1-\alpha}$ for all $\alpha > 0$.

▶ **Proposition 12.** *For each constant $0 < \alpha < 1$, there is a $(1 + \frac{1}{1-\alpha})$-approximation for Directed Optimal Linear Arrangement that runs in $O^*(2^{\frac{\omega n}{3}} + 2^{(1-\frac{\alpha}{4})n})$ time.*

▶ **Proposition 13.** *For each constant $0 < \alpha < 1$, there is a $(1 + \frac{1}{2(1-\alpha)})$-approximation for Optimal Linear Arrangement that runs in $O^*(2^{\frac{\omega n}{3}} + 2^{(1-\frac{\alpha}{4})n})$ time.*

5 Conclusion

In this work, we initiated the study of faster exponential-time approximation algorithms for vertex-ordering problems. For Feedback Arc Set we designed an approximation scheme using a novel self-improving technique. For other problems, we showed how to compute a constant-factor approximation. We mention that the best known polynomial-time algorithms do not achieve a factor better than $O(\sqrt{\log(n)})$.

We conclude with a few open problems. One obvious direction is to design a faster exponential-time algorithm for Directed Minimum $(k, n-k)$-Cut– exact or approximate – which would immediately yield faster approximation algorithms for almost all of the problems considered in this paper. Next, there are a lot more (vertex-ordering) problems that are also interesting to study. Examples include Treewidth, Subset Feedback Vertex/Arc Set (here also the restriction to tournament graphs is often studied), Travelling Sales Person, Bandwidth, or Minimum Fill-In. Second, can our constant-factor approximations be improved to $(1+\varepsilon)$ as in the case of Feedback Arc Set? On a related note, we are not aware of any tool to show lower bounds for exponential-time approximations and most known hypotheses like SETH seem to weak to base such lower bounds on. In particular, a reasonably sounding candidate one could call Gap-SETH (a combination of SETH and Gap-ETH) is not true as shown by Alman et al. [1]. Hence, we ask the following question:

> What would be a sensible hypothesis to exclude $(1+\varepsilon)$-approximations in $O((2-\delta_\varepsilon)^n)$ time for some $\delta_\varepsilon > 0$ depending only on ε for different problems?

Finally, all our algorithms use exponential space (here, the bottleneck is our exact algorithm for Directed Minimum $(k, n-k)$-Cut, which first builds an exponential-size graph). Can this be reduced to polynomial space while still beating the fastest known exact algorithms?

References

1. Josh Alman, Timothy M. Chan, and R. Ryan Williams. Faster deterministic and Las Vegas algorithms for offline approximate nearest neighbors in high dimensions. In *Proceedings of the 31st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 637–649. SIAM, 2020. doi:10.1137/1.9781611975994.39.
2. Andris Ambainis, Kaspars Balodis, Janis Iraids, Martins Kokainis, Krisjanis Prusis, and Jevgenijs Vihrovs. Quantum speedups for exponential-time dynamic programming algorithms. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1783–1793. SIAM, 2019. doi:10.1137/1.9781611975482.107.
3. Per Austrin, Toniann Pitassi, and Yu Wu. Inapproximability of treewidth, one-shot pebbling, and related layout problems. In *Proceedings of the 15th International Workshop on Approximation, Randomization, and Combinatorial Optimization (APPROX)*, pages 13–24. Springer, 2012. doi:10.1007/978-3-642-32512-0_2.
4. Nikhil Bansal, Parinya Chalermsook, Bundit Laekhanukit, Danupon Nanongkai, and Jesper Nederlof. New tools and connections for exponential-time approximation. *Algorithmica*, 81(10):3993–4009, 2019. doi:10.1007/S00453-018-0512-8.
5. Nikhil Bansal, Dor Katzelnick, and Roy Schwartz. On approximating cutwidth and pathwidth. In *Procceedings of the 65th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2024. Accepted for publication. doi:10.1109/FOCS61266.2024.00051.
6. Hans L. Bodlaender, Fedor V. Fomin, Arie M. C. A. Koster, Dieter Kratsch, and Dimitrios M. Thilikos. A note on exact algorithms for vertex ordering problems on graphs. *Theory of Computing Systems*, 50(3):420–432, 2012. doi:10.1007/S00224-011-9312-0.
7. Edouard Bonnet, Bruno Escoffier, Vangelis Th. Paschos, and Emeric Tourniaire. Multi-parameter analysis for local graph partitioning problems: Using greediness for parameterization. *Algorithmica*, 71(3):566–580, 2015. doi:10.1007/S00453-014-9920-6.
8. Nicolas Bourgeois, Bruno Escoffier, and Vangelis Th. Paschos. Approximation of max independent set, min vertex cover and related problems by moderately exponential algorithms. *Discrete Applied Mathematics*, 159(17):1954–1970, 2011. doi:10.1016/J.DAM.2011.07.009.
9. Parinya Chalermsook, Bundit Laekhanukit, and Danupon Nanongkai. Independent set, induced matching, and pricing: Connections and tight (subexponential time) approximation hardnesses. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 370–379. IEEE Computer Society, 2013. doi:10.1109/FOCS.2013.47.
10. Moses Charikar, Mohammad Taghi Hajiaghayi, Howard J. Karloff, and Satish Rao. ℓ_2^2 spreading metrics for vertex ordering problems. *Algorithmica*, 56(4):577–604, 2010.
11. Johanne Cohen, Fedor V. Fomin, Pinar Heggernes, Dieter Kratsch, and Gregory Kucherov. Optimal linear arrangement of interval graphs. In *Proceedings of the 31st International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 267–279. Springer, 2006. doi:10.1007/11821069_24.
12. Marek Cygan, Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. On cutwidth parameterized by vertex cover. *Algorithmica*, 68(4):940–953, 2014. doi:10.1007/S00453-012-9707-6.
13. Marek Cygan and Marcin Pilipczuk. Exact and approximate bandwidth. *Theoretical Computer Science*, 411(40-42):3701–3713, 2010. doi:10.1016/J.TCS.2010.06.018.
14. Irit Dinur and Samuel Safra. On the hardness of approximating minimum vertex cover. *Annals of mathematics*, pages 439–485, 2005.
15. Baris Can Esmer, Ariel Kulik, Dániel Marx, Daniel Neuen, and Roohani Sharma. Faster exponential-time approximation algorithms using approximate monotone local search. In *Proceedings of the 30th Annual European Symposium on Algorithms (ESA)*, pages 50:1–50:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ESA.2022.50.
16. Baris Can Esmer, Ariel Kulik, Dániel Marx, Daniel Neuen, and Roohani Sharma. Approximate monotone local search for weighted problems. In *Proceedings of the 18th International Symposium on Parameterized and Exact Computation (IPEC)*, pages 17:1–17:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.IPEC.2023.17.

17 Baris Can Esmer, Ariel Kulik, Dániel Marx, Daniel Neuen, and Roohani Sharma. Optimally repurposing existing algorithms to obtain exponential-time approximations. In *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 314–345. SIAM, 2024. doi:10.1137/1.9781611977912.13.

18 Guy Even, Joseph Naor, Baruch Schieber, and Madhu Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998. doi:10.1007/PL00009191.

19 Uriel Feige, MohammadTaghi Hajiaghayi, and James R. Lee. Improved approximation algorithms for minimum weight vertex separators. *SIAM Journal on Computing*, 38(2):629–657, 2008. doi:10.1137/05064299X.

20 Uriel Feige and James R. Lee. An improved approximation ratio for the minimum linear arrangement problem. *Information Processing Letters*, 101(1):26–29, 2007. doi:10.1016/J.IPL.2006.07.009.

21 Fedor V. Fomin, Serge Gaspers, Daniel Lokshtanov, and Saket Saurabh. Exact algorithms via monotone local search. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 764–775. ACM, 2016. doi:10.1145/2897518.2897551.

22 Fedor V. Fomin and Dieter Kratsch. *Exact Exponential Algorithms*. Springer, 2010. doi:10.1007/978-3-642-16533-7.

23 Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. In *Proceedings of the 16th ACM national meeting*, page 71. ACM, 1961. doi:10.1145/800029.808532.

24 Thore Husfeldt, Dieter Kratsch, Ramamohan Paturi, and Gregory B. Sorkin, editors. *Exact Complexity of NP-hard Problems*, volume 10441 of *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany, 2010.

25 Thore Husfeldt, Ramamohan Paturi, Gregory B. Sorkin, and Ryan Williams, editors. *Exponential Algorithms: Algorithms and Complexity Beyond Polynomial Time*, volume 13331 of *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany, 2013.

26 Tanmay Inamdar, Madhumita Kundu, Pekka Parviainen, M. S. Ramanujan, and Saket Saurabh. Exponential-time approximation schemes via compression. In *Proceedings of the 15th Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 64:1–64:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.ITCS.2024.64.

27 Alon Itai and Michael Rodeh. Finding a minimum circuit in a graph. *SIAM Journal on Computing*, 7(4):413–423, 1978. doi:10.1137/0207033.

28 Pallavi Jain, Lawqueen Kanesh, William Lochet, Saket Saurabh, and Roohani Sharma. Exact and approximate digraph bandwidth. In *Proceedings of the 39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 18:1–18:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICS.FSTTCS.2019.18.

29 Viggo Kann. *On the approximability of NP-complete optimization problems*. PhD thesis, Royal Institute of Technology Stockholm, 1992.

30 Kenta Kitsunai, Yasuaki Kobayashi, Keita Komuro, Hisao Tamaki, and Toshihiro Tano. Computing directed pathwidth in $O(1.89^n)$ time. *Algorithmica*, 75(1):138–157, 2016.

31 Daniel Lokshtanov, Saket Saurabh, and Vaishali Surianarayanan. Breaking the all subsets barrier for min k-cut. In *Procceedings of the 50th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 90:1–90:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.ICALP.2023.90.

32 Rodica Mihai and Ioan Todinca. Pathwidth is NP-hard for weighted trees. In *Proceedings of the 3rd International Workshop on Frontiers in Algorithmics (FAW)*, pages 181–195. Springer, 2009. doi:10.1007/978-3-642-02270-8_20.

33 Jesper Nederlof. Bipartite TSP in $o(1.9999^n)$ time, assuming quadratic time matrix multiplication. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 40–53. ACM, 2020.

34 Jesper Nederlof, Jakub Pawlewicz, Céline M. F. Swennenhuis, and Karol Wegrzycki. A faster exponential time algorithm for bin packing with a constant number of bins via additive combinatorics. *SIAM Journal of Comput.*, 52(6):1369–1412, 2023. doi:10.1137/22M1478112.

35 Herbert Robbins. A remark on Stirling's formula. *The American Mathematical Monthly*, 62(1):26–29, 1955.

36 Ryan Williams. A new algorithm for optimal constraint satisfaction and its implications. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP)*, pages 1227–1237. Springer, 2004. doi:10.1007/978-3-540-27836-8_101.

37 Or Zamir. Breaking the 2^n barrier for 5-coloring and 6-coloring. In *Procceedings of the 48th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 113:1–113:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.

38 Uri Zwick. All pairs shortest paths using bridging sets and rectangular matrix multiplication. *Journal of the ACM*, 49(3):289–317, 2002. doi:10.1145/567112.567114.

On Fault Tolerant Single-Shot Logical State Preparation and Robust Long-Range Entanglement

Thiago Bergamaschi
University of California Berkeley, CA, USA

Yunchao Liu
University of California Berkeley, CA, USA
Harvard University, Cambridge, MA, USA

Abstract

Preparing encoded logical states is the first step in a fault-tolerant quantum computation. Standard approaches based on concatenation or repeated measurement incur a significant time overhead. The Raussendorf-Bravyi-Harrington cluster state [23] offers an alternative: a single-shot preparation of encoded states of the surface code, by means of a constant depth quantum circuit, followed by a single round of measurement and classical feedforward [7]. In this work we generalize this approach and prove that single-shot logical state preparation can be achieved for arbitrary quantum LDPC codes. Our proof relies on a minimum-weight decoder and is based on a generalization of Gottesman's clustering-of-errors argument [12]. As an application, we also prove single-shot preparation of the encoded GHZ state in arbitrary quantum LDPC codes. This shows that adaptive noisy constant depth quantum circuits are capable of generating generic robust long-range entanglement.

2012 ACM Subject Classification Theory of computation → Quantum computation theory

Keywords and phrases Quantum error correction, fault tolerance, single-shot error correction, logical state preparation

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.16

Related Version *Full Version*: https://arxiv.org/abs/2411.04405 [2]

Funding T.B. acknowledges support by the National Science Foundation under Grant No. DGE 2146752. Y.L. is supported by DOE Grant No. DE-SC0024124, NSF Grant No. 2311733, and DOE Quantum Systems Accelerator.

Acknowledgements We thank Zhiyang He (Sunny) for helpful discussions. We thank Jong Yeon Lee for helpful discussions and for informing us about an upcoming work with Isaac Kim on single-shot error correction and cluster states. This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing.

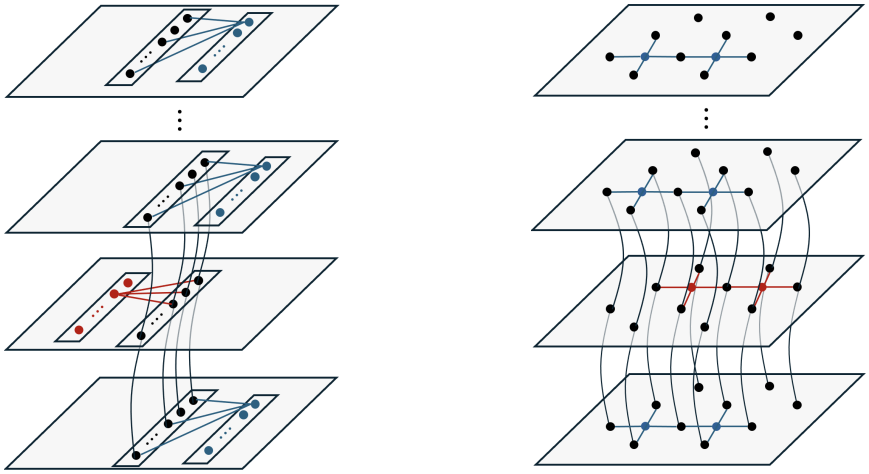
1 Introduction

Single-shot logical state preparation is a procedure in which a code state (such as the logical $|0\rangle$ or $|+\rangle$ state) of a quantum error correcting code is prepared by a constant depth quantum circuit, followed by one round of single qubit measurements, and an adaptive Pauli correction using classical feedforward. In this paper, we prove that every CSS quantum LDPC code admits a single-shot state preparation procedure with logarithmic space overhead that is fault tolerant against local stochastic noise, generalizing prior work by [7] for the surface code.

The fact that single-shot logical state preparation is feasible is non-trivial: after all, logical states of quantum LDPC codes have circuit lower bounds and cannot be prepared by a constant depth quantum circuit [6, 14, 1]. This issue is circumvented via the additional measurement and classical feedforward. Due to the simplicity of the quantum circuit, there has been a recent theoretical and experimental interest in preparing physically interesting


© Thiago Bergamaschi and Yunchao Liu;
licensed under Creative Commons License CC-BY 4.0





(a) The Alternating Tanner Graph state $|\mathrm{ATG}\rangle$. **(b)** $|\mathrm{ATG}\rangle$ of the surface code.

Figure 1 (a) The ATG is a graph state, defined on a vertex set comprised of layers of copies of the code block and the Z (resp. X) Tanner graph (in blue, resp. red) of the qLDPC code. (b) The RBH cluster state [23], which prepares code states of the surface code in a 3D cubic lattice arrangement, is a special case of this construction.

quantum states using these adaptive shallow circuits [22, 20, 21, 8, 25, 10, 28, 16, 24]. Fault tolerance is a key aspect of a faithful experimental preparation of these states, where one must argue that the state is still interesting even though the shallow circuit is noisy. Our result provides a general recipe for fault tolerant state preparation, which can be used for state initialization in fault tolerant quantum computation, as well as a candidate for an experimental demonstration of long-range entangled states using adaptive shallow circuits that is robust to noise.

The intuition behind our result is to realize *repeated measurement* via *measurement-based quantum computation*. The standard approach to initialize a quantum LDPC code in the logical $|+\rangle$ state is as follows: first initialize all physical qubits in a code block in the $|+\rangle$ state, then perform repeated Z and X syndrome measurements, and finally perform a Pauli error correction using classical feedforward based on the syndrome measurement outcomes. A key observation here is that the repeated measurements are non-adaptive, thus we can attempt to simulate this process using a cluster state, achieving a space-time trade-off.

This naturally leads to a construction which we refer to as the Alternating Tanner Graph state $|\mathrm{ATG}\rangle$ (Figure 1, also known as a foliated quantum code [3]): there are $2T+1$ copies of the code block. A copy of the Z Tanner graph is placed on each odd layer (simulating Z syndrome measurements), and a copy of the X Tanner graph is placed on each even layer (simulating X syndrome measurements). Vertical connections are added between code qubits of neighboring layers, which is used to propagate quantum computation in the time direction. To prepare this cluster state in constant depth, we initialize all qubits in the $|+\rangle$ state and apply a CZ gate on each edge. When applied to the surface code, the $|\mathrm{ATG}\rangle$ is equivalent to the Raussendorf-Bravyi-Harrington (RBH) cluster state [23].

1.1 Results

We start with an informal description of single-shot logical state preparation. Consider a set of noisy qubits divided into two subsystems A and B. Single-shot logical state preparation is performed via the following procedure:

1. Initialize all qubits in the (noisy) $|+\rangle$ state.
2. Perform a (noisy) constant depth quantum circuit W.
3. Measure all qubits in A in the Hadamard $(|+\rangle, |-\rangle)$ basis, obtaining a bitstring $s \in \{0,1\}^{|A|}$. In the absence of noise, the unnormalized post-measurement state equals $|\pm_s\rangle\langle\pm_s|_A \cdot W |+\rangle_{AB}$.
4. Use a classical computer to calculate a Pauli operator $P(s)$ (supported on B) based on s, and apply $P(s)$ to the post-measurement state.[1]

In the absence of noise, the final (unnormalized) state is given by

$$\left(|\pm_s\rangle\langle\pm_s|_A \otimes P(s)_B\right) \cdot W |+\rangle_{AB} . \tag{1}$$

Let us begin by understanding what this procedure can achieve in the absence of noise. With a suitable choice of W, one can imagine that step 3 is measuring the stabilizers/parity checks of a quantum LDPC code (where B contains code qubits and A contains ancilla qubits for syndrome measurement), which collapses the state into a random syndrome subspace s (some eigenspace of the code Hamiltonian). Step 4 then performs syndrome decoding and corrects the Pauli error, thus preparing an encoded code state.[2]

The key question is what can the above procedure achieve in the noisy setting, where every initial qubit, gate and measurement is subject to some small constant amount of noise (throughout this section, we assume the noise rate is below some constant threshold). Ideally, we would like to achieve *fault-tolerant* single-shot logical state preparation, which means that the final state in the presence of noise is the desired encoded logical state, up to some residual stochastic noise. This residual noise is benign in the sense that it is correctable with the LDPC code; and the final state (with residual noise) can be used as the initial state of a fault tolerant quantum computation.

The conceptual challenge to achieve fault tolerance here is that the noisy quantum circuit only has constant depth, therefore we cannot hope to correct errors during the state preparation procedure using standard means. The RBH cluster state and its generalization, the ATG [23, 3, 15], is a carefully designed construction where after the bulk of the state is measured, the remaining boundaries/surfaces encode a logical state in the quantum error correcting code up to a Pauli correction. The intuition that this scheme might be fault tolerant lies in the redundancy embedded into the bulk ancilla qubits. Here, we give a general fault-tolerance proof that works for arbitrary quantum LDPC codes.

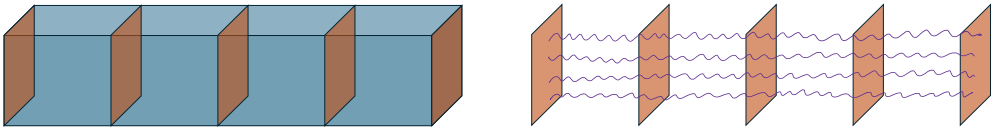
▶ **Theorem 1.** *Fix an integer T. There exists a fault tolerant, single shot logical state preparation procedure for the encoded $|\overline{0}\rangle, |\overline{+}\rangle$ states[3] of an arbitrary $[[n,k,d]]$ CSS LDPC code, using $O(n \cdot T)$ ancilla qubits, with success probability at least $1 - nT \cdot 2^{-\Omega(\min(d,T))}$.*

As discussed earlier, the Alternating Tanner Graph (ATG) state (Figure 1) naturally gives rise to redundancies akin to those in a repeated syndrome measurements protocol. Our proof of fault-tolerance leverages an information-theoretic minimum-weight decoder, and is based on the clustering-of-errors argument by [12, 18] for decoding LDPC codes from repeated measurements (see below for a more detailed overview).

[1] Classical computation is assumed to be performed instantly, such that there is no additional noise on the quantum state during this step of the computation.

[2] What code state are we preparing? For example, assume there is only one logical qubit. Notice that the logical X operator of the CSS LDPC code is stabilized by the initial state and commutes with the measurement. Therefore we are able to prepare the encoded $|+\rangle$ state. The encoded $|0\rangle$ state can be prepared similarly by initializing all qubits in $|0\rangle$.

[3] Note that there are k logical qubits, and we can prepare either $\left|\overline{0}^{\otimes k}\right\rangle$ or $\left|\overline{+}^{\otimes k}\right\rangle$.



(a) Measurement pattern of $|\mathrm{ATG}\rangle$. **(b)** Encoded GHZ state.

Figure 2 (a) We prepare $|\mathrm{ATG}\rangle$ of an arbitrary LDPC code using a noisy shallow circuit and measure the bulk (blue), leaving m surfaces (orange) unmeasured. (b) After decoding and feedforward, the state becomes the encoded GHZ state across the m surfaces, up to residual stochastic noise.

Efficient decoding. In our single-shot logical state preparation result discussed above, the classical decoding procedure (computing $P(s)$ from s) relies on an information-theoretic minimum weight decoder, and is not necessarily computationally efficient.[4] A natural question is whether an efficient decoder that runs in $\mathsf{poly}(n)$ time exists for single-shot logical state preparation, especially if the code is known to have an efficient decoder for quantum error correction.[5] Here, we complement our results by showing a simple reduction to the efficient decoding of logical state preparation based on repeated syndrome measurements.

▶ **Theorem 2.** *If a family of $[[n, k, d]]$ CSS LDPC codes admits an efficient decoder for fault-tolerant logical state preparation based on T rounds of repeated syndrome measurements, then it further admits an efficient decoder for a fault-tolerant single shot logical state preparation procedure using $O(n \cdot T)$ ancilla qubits.*

In other words, one can tradeoff time for space in logical state preparation. As discussed earlier, this reduction stems from mapping the repeated measurements circuit to a constant depth circuit via measurement based quantum computation, and as we illustrate naturally gives rise to the alternating tanner graph state $|\mathrm{ATG}\rangle$.

Robust long-range entanglement. Next we show that the ATG provides a powerful framework that goes beyond the preparation of encoded $|\overline{0}\rangle$ and $|\overline{+}\rangle$ as in logical state preparation based on repeated measurements, due to the flexibility in *choosing different measurement patterns* in the ATG. As an example, we show fault tolerant single-shot preparation of the encoded GHZ state

$$|\mathrm{GHZ}_m\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes m} + |1\rangle^{\otimes m}\right) \tag{2}$$

for any $m \geq 2$ in an arbitrary quantum LDPC code. Here we summarize the algorithm for GHZ state preparation, and then discuss this result from the perspective of robust long-range entanglement and noisy shallow circuits.

To prepare the GHZ state, we measure the ATG in the following pattern, depicted (horizontally) in Figure 2a: we measure all the check qubits of all code blocks (blue and red in Figure 1), as well as all code qubits in the bulk (blue in Figure 2a), with the exception of a careful choice of m layers of n code qubits (in orange). We prove that the resulting state, after minimum-weight decoding and feedforward, is in fact the encoded GHZ state across the m blocks (Figure 2b) up to residual stochastic noise.

[4] By dynamic programming, the decoding algorithm can be made to run in $\leq T \cdot \exp(n)$ time.
[5] We emphasize the distinction between a decoder for error-correction, and a decoder for state preparation.

▶ **Theorem 3.** *Fix integers m, T. There exists a fault tolerant, single shot logical state preparation procedure for the encoded $\left|\overline{\mathrm{GHZ}_m^{\otimes k}}\right\rangle$ state of an arbitrary $[[n, k, d]]$ CSS LDPC code, using $O(n \cdot T)$ ancilla qubits, with success probability at least $1 - nT \cdot 2^{-\Omega(\min(d, T/m))}$.*

This generalizes the RBH construction [23] for the surface code with $m = 2$ and $k = 1$, that is, the encoded Bell state $\frac{1}{\sqrt{2}}\left(\left|\overline{00}\right\rangle + \left|\overline{11}\right\rangle\right)$ in the surface code. Let us review the features of the entanglement in the resulting GHZ state:

- **Noise robust.** The preparation procedure uses a noisy constant depth circuit (with noise rate below a constant threshold). Yet, the resulting GHZ state is encoded in a LDPC code and is subject to residual local stochastic noise that is correctable with the code.
- **Generic.** The construction works for arbitrary quantum LDPC codes.
- **Macroscopic.** The final state encodes $m \cdot k$ logical qubits into $m \cdot n$ physical qubits, which can achieve a constant encoding rate using high-rate codes such as hypergraph product codes [26, 17].
- **Long range.** There are two aspects of long range entanglement: the first is the entanglement among physical qubits in each code block (which can exhibit topological order), the second is the logical GHZ entanglement that spans m code blocks.

Finally, we discuss this result from the perspective of noisy shallow quantum circuits. The first step of this state preparation algorithm is to apply a constant depth circuit on a volume of qubits. At this point, there is no long-range entanglement at all. However, after the bulk (blue in Figure 2a) is measured, long-range entanglement is created across the orange blocks (Figure 2b): the subsequent decoding and feedforward operations are only used to correct the state into the desired GHZ state, but cannot create entanglement. Therefore, the GHZ-type entanglement is created by the teleportation effect of measurement. Interestingly, this teleportation effect is robust to noise.

We remark that Theorem 3 implies Theorem 1 as a special case. An interesting future direction is to explore whether there exists other families of interesting states that can be prepared using the ATG framework.

Proof of fault tolerance. The fault-tolerance of our single-shot logical state preparation scheme has two main components. First, we identify the stabilizer structure of $|\mathrm{ATG}\rangle$. As remarked earlier, the fault tolerance arises from redundancies in the ancilla qubits. Concretely, that manifests in the redundancies of the stabilizers of $|\mathrm{ATG}\rangle$, which contains certain "meta-checks". Second, we adapt the clustering arguments by [18, 12] for decoding quantum LDPC codes from random errors to our setting, and develop analogous techniques to argue that the meta-checks contain enough redundancy to be robust against random errors.

The meta-check information. Following [7], inside the $|\mathrm{ATG}\rangle$ (which is a stabilizer state), one can define stabilizers which cleanly factorize into a product of X Pauli's on the bulk of the graph state. We refer to these stabilizers as "meta-checks", since

1. After measuring the bulk in the X basis, they remain stabilizers of the post-measurement state,

2. They inbuild redundancies ("syndrome of syndromes") into the measured string s.

These redundancies will later allow us to use s to infer errors that occur on the bulk, and in turn, correctly decode the error $P(s)$ for the qubits on the boundary. Here we give an example and defer a rigorous exposition to [2, Section 3.2]. Recall that on every even layer t

of the ATG, there is a copy of the Z-Tanner graph lying on the layers $t-1$ and $t+1$ adjacent to it. Now, consider a Z-check of the code denoted as c, and let $\mathsf{Supp}(c) \subseteq [n]$ be the set of qubits it acts on. We show that the following operator

$$X_{(c,t-1)} \otimes X_{(c,t+1)} \bigotimes_{i \in \mathsf{Supp}(c)} X_{(i,t)}, \tag{3}$$

which applies a Pauli-X on the copies of the check qubit c in layers $t-1$ and $t+1$, as well as the qubits supported in the check in layer t, is a stabilizer of $|\mathrm{ATG}\rangle$.

While perhaps a bit abstract at first, this operator captures the change in the Z syndrome after a layer of data-errors on the code and measurement errors on the checks: akin to the redundancy between adjacent syndrome measurements that appear in a repeated syndrome measurement protocol. The fact that these stabilizers factorize cleanly into products of X Paulis is non-trivial and depends carefully on the structure of $|\mathrm{ATG}\rangle$, which we discuss further in [2, Section 3.2].

The clustering argument. [18, 12] showed that any quantum LDPC code can correct random Pauli errors (below a constant threshold noise rate), even in the presence of syndrome measurement errors, by repeating the syndrome measurements T times and running an information-theoretic minimum weight decoder. Their proof reasoned that the data-errors and syndrome-errors could be arranged in a low degree graph known as the *syndrome adjacency graph*, wherein the locations of mismatch errors (where the output of the decoder does not match the true error) clusters into connected components. By a percolation argument on this low-degree graph, one can argue these clusters are unlikely to "contrive" into configurations that induce logical errors on the code. Roughly speaking, the probability of a logical error is upper bounded by

$$\begin{aligned} &\sum_{s \geq d} \underbrace{\left(\#\text{ of Connected Components } |K| = s\right)}_{\text{A union bound over clusters of size } s} \times \underbrace{\left(p^{\Omega(s)}\right)}_{\text{Odds of such a cluster}} \\ &\approx \sum_{s \geq d} (p \cdot z)^{\Omega(s)} \approx (pz)^d, \end{aligned} \tag{4}$$

where only clusters of size larger than the code distance d can incur logical errors, and z is the degree of the graph.[6]

Here, we broadly follow this strategy and construct an analogous low-degree syndrome adjacency graph which captures where our min-weight decoder incorrectly guesses a data or check error. Unfortunately, making this argument rigorous is easier said than done. Arguably the main challenge lies in relating the size of clusters of mismatch errors, to the number of true physical errors which lie within the cluster, to properly claim the decay rate $p^{\Omega(s)}$ with the cluster size. As we discuss in [2, Section 5] and [2, Appendix C], this calculation is highly dependent on the boundary conditions of the syndrome adjacency graph, tailored for Bell states and to GHZ states.

1.2 Discussion and other related works

Fault tolerant shallow circuits. As a direct corollary, our single-shot logical state preparation can be used as the basis of a fault tolerant implementation of a shallow quantum circuit.

[6] A function only of the locality of the LDPC code. We assumed the noise rate is bounded by a threshold $p < z^{-1}$ for the geometric series to converge.

▶ **Corollary 4.** *Fix an arbitrary quantum LDPC code Q. Let C be a constant depth quantum circuit which uses $|0\rangle$ or $|+\rangle$ or GHZ initial states, gates which are transversal in Q, and measurements in the standard or Hadamard basis. Then there is a constant depth fault tolerant implementation of C which uses a single round of mid-circuit measurement and feedforward.*

Here we simply run our single-shot logical state preparation scheme to prepare encoded initial states, and then directly apply transversal gates. Since the logical circuit is constant depth, the residual local stochastic noise still remains local stochastic, and the logical Z or X measurements at the end can be decoded using the code.

A natural question is whether we can remove the mid-circuit measurement and feedforward, and implement an ideal shallow quantum circuit purely via a noisy shallow quantum circuit. Ref. [7] gives such an example: starting from RBH, they directly apply a logical Clifford circuit *without* correcting the Pauli error $P(s)$. However, since the logical circuit is Clifford, the Pauli error $P(s)$ is propagated through the circuit and corrected at the end. Our result similarly implies a more general version:

▶ **Corollary 5.** *Fix an arbitrary quantum LDPC code Q. Let C be a constant depth quantum circuit which uses $|0\rangle$ or $|+\rangle$ or GHZ initial states, and gates which are Clifford and transversal in Q, and measurements in the standard or Hadamard basis. Then there is a constant depth fault tolerant implementation of C using a noisy circuit (without mid-circuit measurements and feedforward).*

Note that we do not address the issue of decoding efficiency. In particular, although Corollary 5 does not need mid-circuit decoding, the decoding at the end of the circuit may still be inefficient. An interesting direction is to explore if Corollary 5 can be the basis of other quantum advantage experiments with noisy shallow quantum circuits against shallow classical circuits, as in [7].

Single-shot preparation of general stabilizer states. Beyond GHZ state preparation, we show that single-shot logical state preparation can also be achieved for more general stabilizer states, using a combination of the ATG and Steane's logical measurement. Let $|\psi\rangle$ be a m-qubit stabilizer state with the following structure: there exists a set of m stabilizer generators that can be divided into two sets $S_X \subseteq \{I, X\}^{\otimes m}$ and $S_Z \subseteq \{I, Z\}^{\otimes m}$, such that either S_X or S_Z has constant weight and degree. The m-qubit GHZ state is such an example, where $S_X = \{X^{\otimes m}\}$ and $S_Z = \{Z_1Z_2, Z_2Z_3, \ldots, Z_{m-1}Z_m\}$. We prove fault tolerant single-shot preparation for any such state $|\psi\rangle$ encoded across m copies of an arbitrary quantum LDPC code. See [2, Appendix D] for more details.

Single-shot quantum error correction. A closely related concept to our work is single-shot quantum error correction [5]: consider an encoded code block subject to random errors, it is shown that for certain codes a single round of syndrome measurement suffices to correct the error (instead of performing repeated syndrome measurements), even in the presence of syndrome measurement errors. Examples include high-dimensional color codes [4, 5] and toric codes [19], hypergraph product codes [11], quantum Tanner codes [13] and hyperbolic codes [9].

As the ATG is closely related to repeated syndrome measurements, it is natural to ask whether single-shot logical state preparation with *constant* space overhead exists for those codes, or more generally, for any code with single-shot error correction. While this is proven for 3D gauge color codes [4, 5], it is unclear whether a general reduction from single-shot

logical state preparation to single-shot error correction exists. The key issue is that single-shot error correction is defined in the context of correcting errors on an existing encoded code block, while single-shot logical state preparation is about preparing an encoded code state *from scratch*. We leave further exploration of this issue for future work; it is an interesting question in the context of reducing the spacetime overhead of quantum fault tolerance (also see [27]).

References

1 Dorit Aharonov and Yonathan Touati. Quantum circuit depth lower bounds for homological codes, 2018. arXiv:1810.03912.

2 Thiago Bergamaschi and Yunchao Liu. On fault tolerant single-shot logical state preparation and robust long-range entanglement, 2024. arXiv:2411.04405.

3 A. Bolt, G. Duclos-Cianci, D. Poulin, and T. M. Stace. Foliated quantum error-correcting codes. *Phys. Rev. Lett.*, 117:070501, August 2016. doi:10.1103/PhysRevLett.117.070501.

4 H. Bombin. Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes. *New Journal of Physics*, 17, 2013. URL: https://api.semanticscholar.org/CorpusID:3355844.

5 Héctor Bombín. Single-shot fault-tolerant quantum error correction. *Phys. Rev. X*, 5:031043, September 2015. doi:10.1103/PhysRevX.5.031043.

6 S. Bravyi, M. B. Hastings, and F. Verstraete. Lieb-robinson bounds and the generation of correlations and topological quantum order. *Phys. Rev. Lett.*, 97:050401, July 2006. doi:10.1103/PhysRevLett.97.050401.

7 Sergey Bravyi, David Gosset, Robert König, and Marco Tomamichel. Quantum advantage with noisy shallow circuits. *Nature Physics*, 16(10):1040–1045, October 2020. doi:10.1038/s41567-020-0948-z.

8 Sergey Bravyi, Isaac Kim, Alexander Kliesch, and Robert Koenig. Adaptive constant-depth circuits for manipulating non-abelian anyons, 2022. arXiv:2205.01933.

9 N. P. Breuckmann and Vivien Londe. Single-shot decoding of linear rate ldpc quantum codes with high performance. *IEEE Transactions on Information Theory*, 68:272–286, 2020. URL: https://api.semanticscholar.org/CorpusID:210157092.

10 Edward H. Chen, Guo-Yi Zhu, Ruben Verresen, Alireza Seif, Elisa Bäumer, David Layden, Nathanan Tantivasadakarn, Guanyu Zhu, Sarah Sheldon, Ashvin Vishwanath, Simon Trebst, and Abhinav Kandala. Realizing the nishimori transition across the error threshold for constant-depth quantum circuits, 2023. arXiv:2309.02863.

11 Omar Fawzi, Antoine Grospellier, and Anthony Leverrier. Efficient decoding of random errors for quantum expander codes. *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 2017. URL: https://api.semanticscholar.org/CorpusID:4390794.

12 Daniel Gottesman. Fault-tolerant quantum computation with constant overhead. *Quantum Info. Comput.*, 14(15–16):1338–1372, November 2014. doi:10.26421/QIC14.15-16-5.

13 Shouzhen Gu, Eugene Tang, Libor Caha, Shin Ho Choe, Zhiyang He, and Aleksander Kubica. Single-shot decoding of good quantum ldpc codes. *ArXiv*, abs/2306.12470, 2023. doi:10.48550/arXiv.2306.12470.

14 Jeongwan Haah. An invariant of topologically ordered states under local unitary transformations. *Communications in Mathematical Physics*, 342(3):771–801, March 2016. doi:10.1007/s00220-016-2594-y.

15 Timo Hillmann, Guillaume Dauphinais, Ilan Tzitrin, and Michael Vasmer. Single-shot and measurement-based quantum error correction via fault complexes, 2024. arXiv:2410.12963.

16 Mohsin Iqbal, Nathanan Tantivasadakarn, Thomas M. Gatterman, Justin A. Gerber, Kevin Gilmore, Dan Gresh, Aaron Hankin, Nathan Hewitt, Chandler V. Horst, Mitchell Matheny, Tanner Mengle, Brian Neyenhuis, Ashvin Vishwanath, Michael Foss-Feig, Ruben Verresen, and Henrik Dreyer. Topological order from measurements and feed-forward on a trapped ion quantum computer. *Communications Physics*, 7(1):205, June 2024. doi:10.1038/s42005-024-01698-3.

17 Alexey A. Kovalev and Leonid P. Pryadko. Improved quantum hypergraph-product ldpc codes. In *2012 IEEE International Symposium on Information Theory Proceedings*, pages 348–352, 2012. doi:10.1109/ISIT.2012.6284206.

18 Alexey A. Kovalev and Leonid P. Pryadko. Fault tolerance of quantum low-density parity check codes with sublinear distance scaling. *Phys. Rev. A*, 87:020304, February 2013. doi:10.1103/PhysRevA.87.020304.

19 Aleksander Kubica and Michael Vasmer. Single-shot quantum error correction with the three-dimensional subsystem toric code. *Nature Communications*, 13, 2021. URL: https://api.semanticscholar.org/CorpusID:235352916.

20 Jong Yeon Lee, Wenjie Ji, Zhen Bi, and Matthew P. A. Fisher. Decoding measurement-prepared quantum phases and transitions: from ising model to gauge theory, and beyond, 2022. arXiv:2208.11699.

21 Jong Yeon Lee, Yi-Zhuang You, and Cenke Xu. Symmetry protected topological phases under decoherence, 2024. arXiv:2210.16323.

22 Tsung-Cheng Lu, Leonardo A. Lessa, Isaac H. Kim, and Timothy H. Hsieh. Measurement as a shortcut to long-range entangled quantum matter. *PRX Quantum*, 3:040337, December 2022. doi:10.1103/PRXQuantum.3.040337.

23 Robert Raussendorf, Sergey Bravyi, and Jim Harrington. Long-range quantum entanglement in noisy cluster states. *Phys. Rev. A*, 71:062313, June 2005. doi:10.1103/PhysRevA.71.062313.

24 Nathanan Tantivasadakarn, Ryan Thorngren, Ashvin Vishwanath, and Ruben Verresen. Long-range entanglement from measuring symmetry-protected topological phases. *Phys. Rev. X*, 14:021040, June 2024. doi:10.1103/PhysRevX.14.021040.

25 Nathanan Tantivasadakarn, Ashvin Vishwanath, and Ruben Verresen. Hierarchy of topological order from finite-depth unitaries, measurement, and feedforward. *PRX Quantum*, 4:020339, June 2023. doi:10.1103/PRXQuantum.4.020339.

26 Jean-Pierre Tillich and Gilles Zémor. Quantum ldpc codes with positive rate and minimum distance proportional to the square root of the blocklength. *IEEE Transactions on Information Theory*, 60(2):1193–1202, 2014. doi:10.1109/TIT.2013.2292061.

27 Hengyun Zhou, Chen Zhao, Madelyn Cain, Dolev Bluvstein, Casey Duckering, Hong-Ye Hu, Sheng-Tao Wang, Aleksander Kubica, and Mikhail D. Lukin. Algorithmic fault tolerance for fast quantum computing, 2024. arXiv:2406.17653.

28 Guo-Yi Zhu, Nathanan Tantivasadakarn, Ashvin Vishwanath, Simon Trebst, and Ruben Verresen. Nishimori's cat: Stable long-range entanglement from finite-depth unitaries and weak measurements. *Phys. Rev. Lett.*, 131:200201, November 2023. doi:10.1103/PhysRevLett.131.200201.

Random Restrictions of Bounded Low Degree Polynomials Are Juntas

Sreejata Kishor Bhattacharya
School of Technology and Computer Science, Tata Institute of Fundamental Research, Mumbai, India

Abstract

We study the effects of random restrictions on low degree functions that are *bounded* on every point of the Boolean cube. Our main result shows that, with high probability, the restricted function can be approximated by a junta of arity that is just polynomial in the original degree. More precisely, let $f : \{\pm 1\}^n \to [0, 1]$ be a degree d polynomial ($d \geq 2$) and let ρ denote a random restriction with survival probability $O(\log(d)/d)$. Then, with probability at least $1 - d^{-\Omega(1)}$, there exists a function $g : \{\pm 1\}^n \to [0, 1]$ depending on at most $d^{O(1)}$ coordinates such that $||f_\rho - g||_2^2 \leq d^{-1-\Omega(1)}$.

Our result has the following consequence for the well known, outstanding conjecture of Aaronson and Ambainis. The Aaronson-Ambainis conjecture was formulated to show that the acceptance probability of a quantum query algorithm can be well approximated almost everywhere by a classical query algorithm with a polynomial blow-up: it speculates that a polynomal $f : \{\pm 1\}^n \to [0, 1]$ with degree d has a coordinate with influence $\geq \text{poly}(1/d, \mathsf{Var}[f])$. Our result shows that this is true for a non-negligible fraction of random restrictions of f assuming $\mathsf{Var}[f]$ is not too low.

Our work combines the ideas of Dinur, Friedgut, Kindler and O'Donnell [7] with an approximation theoretic result, first reported in the recent work of Filmus, Hatami, Keller and Lifshitz [8].

2012 ACM Subject Classification Theory of computation → Quantum query complexity

Keywords and phrases Analysis of Boolean Functions, Quantum Query Algorithms

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.17

Related Version *Full Version*: https://arxiv.org/abs/2402.13952

Funding *Sreejata Kishor Bhattacharya*: Google Fellowship 2024.

Acknowledgements I want to thank Francisco Escudero Gutierrez for pointing out typos and a small mistake in a previous version of this paper.

1 Introduction

The analysis of Boolean functions is a vibrant field with connections and applications to diverse areas. While several outstanding results have been established for Boolean functions, the closely related class of *bounded* functions, i.e. those which take value in the real interval $[0, 1]$ at every point of the Boolean cube, remains far less understood. Indeed, such bounded functions are of great significance. As mentioned in Dinur, Friedgut, Kindler and O'Donnell [7], "such bounded functions often arise naturally, particularly as weighted averages of boolean functions; e.g., as Fourier transforms of boolean functions, as noise-convolutions of boolean functions, in the context of random walks on the discrete cube, and in hardness-of approximation theory in computational complexity", and it is often important to understand when such functions depend essentially only on a few number of coordinates.

In this work, we study the effects of random restrictions on such bounded functions of low degree. The study of random restrictions of Boolean functions has been invaluable for establishing circuit lower bounds (see for example, [9, 10, 21]), proof complexity (see for example, [5, 11]), de-randomization (see for example, [12]), and several other areas. Our main result shows a remarkable similarity in the behavior of bounded functions with their Boolean cousins, when hit with random restrictions.

© Sreejata Kishor Bhattacharya;
licensed under Creative Commons License CC-BY 4.0

By now, it is worth recalling that a classical result of Dinur, Friedgut, Kindler and O'Donnell [7] showed that bounded functions whose Fourier tail above degree k is $\leq \exp(-O(k^2))$ are approximable by juntas with arity $\exp(O(k))$. They also showed that this result is tight in general. We are interested in the case when the function has small degree d. What additional structure follows from this stronger condition? To take cue on what to expect, let's recall the case of Boolean functions. If $f : \{\pm 1\}^n \to \{0,1\}$ is a degree d Boolean function, a seminal result of Beals, Buhrman, Cleve, Mosca, and de Wolf [4] shows that f can be represented by a DNF of width $d^{O(1)}$. By the celebrated Hastad's switching lemma, this implies that if f is hit with a random restriction with survival probability $1/d^{O(1)}$, with high probability f becomes constant. We show something similar is true for bounded functions as well: if $f : \{\pm 1\}^n \to [0,1]$ is a degree d, *bounded* function and f is hit with a random restriction with survival probability $O(\log(d)/d)$, with high probability it can be well-approximated by a junta depending only on $\text{poly}(d)$ coordinates. This is despite the fact that the expected number of variables that remain alive is $O\left(n \cdot \frac{\log d}{d}\right)$.

▶ **Theorem 1 (Main Theorem).** *For any constants $C_1, C_2 > 0$, there exist constants $C_3, C_4, C_5 > 0$ such that the following holds:*
Let $f : \{\pm 1\}^n \to [0,1]$ be a degree d polynomial and let ρ be a random restriction with survival probability $\frac{\log(d)}{C_3 d}$. With probability at least $1 - d^{-C_2}$, f_ρ is a $(d^{-C_1}\mathsf{Var}[f], \mathsf{Var}[f]^{-2}d^{C_4})$ junta. Moreover, if J denotes the set of coordinates on which the junta depends, for each $j \in J$ we have $\mathsf{Inf}_j[f] \geq \mathsf{Var}[f]^2 d^{-C_5}$.

This result has interesting consequences for the Aaronson-Ambainis conjecture, which we explain next.

One of the central open problems in the field of quantum query complexity is determining if there exists a partial function which is defined on a large fraction of the Boolean hypercube (say, constant) but whose quantum query complexity and classical query complexity are super-polynomially separated. The result of Beals et. al. [4] shows that no such separation is possible when the function is defined on the entire hypercube. On the other hand, functions for which we know such a separation (e.g. - Forrelation [1] , Bernstein-Vazirani [6]) are defined on an exponentially small fraction of the hypercube. A possible explanation as to why all known functions exhibiting large gaps between quantum and classical query complexity have very small support size would be the following folklore conjecture:

▶ **Conjecture 2.** *Let Q be a quantum query algorithm with Boolean output on n qubits making q queries. Let $P : \{\pm 1\}^n \to [0,1]$ be given by $P(x) = Pr[Q \text{ outputs 1 on } x]$. For any $\epsilon > 0$, there exists a classical query algorithm A such that $\mathsf{E}[(A(x) - Q(x))^2] \leq \epsilon$ and A makes at most $\mathsf{poly}\left(q, \frac{1}{\epsilon}\right)$ queries.*

It is known that if Q makes at most q queries, then P is given by a polynomial of degree at most $2q$. Although P has more structure than any arbitrary low degree bounded polynomial, it is further conjectured that such structure is not necessary. In other words, we forget the fact that P arises from a quantum query algorithm and instead try to construct a classical query algorithm for *any* bounded low-degree polynomial. This led to the following conjecture (also folklore).

▶ **Conjecture 3.** *Let $P : \{\pm 1\}^n \to [0,1]$ be a degree d polynomial. For any $\epsilon > 0$, there exists a classical decision tree T of depth at most $\mathsf{poly}(d, 1/\epsilon)$ such that $\mathsf{E}[(P(x) - T(x))^2] \leq \epsilon$.*

Aaronson and Ambainis [2] proposed the following query algorithm to estimate P: suppose the variance of the function is sufficiently small. Then we terminate the query algorithm and output the average over the unqueried coordinates. If not, we query the coordinate with the highest *influence* and restrict the function according to the response received. We keep doing this until we have made too many queries or the variance has become sufficiently low. In order to show that this algorithm gives an accurate estimate, [2] observed that it is sufficient to prove the following conjecture.

▶ **Conjecture 4** (Aaronson-Ambainis conjecture). *Let $f : \{\pm 1\}^n \to [0,1]$ be a degree d polynomial. Then, there exists a coordinate j such that $\mathsf{Inf}_j[f] \geq \mathsf{poly}(1/d, \mathsf{Var}[f])$.*

As a side remark, we mention that O'Donnell et al. [18] had shown previously that functions which can be approximated by decision trees have a coordinate with high influence. So conjectures 3 and 4 are equivalent.

The Aaronson-Ambainis Conjecture has received significant attention in the past few years. In 2006, a result of Dinur, Friedgut, Kindler and O'Donnell [7] showed that the conjecture is true if $\mathsf{poly}(d)$ is replaced by $\exp(d)$. In 2012, Montanaro [16] proved the conjecture in the special case of block-multilinear forms where all coefficients have the same magnitude. In 2016, O'Donnell and Zhao [19] showed that it suffices to prove the conjecture for a special class of polynomials known as *one-block decoupled polynomials*. In 2020, Keller and Klein [13] claimed to have found a proof for the conjecture but their paper had a subtle flaw and turned out to be wrong. More recently, Lovett and Zhang [15] initiated a new line of attack using the notions of *fractional block sensitivity* and *fractional certificate complexity*. In 2022, Bansal, Sinha and de Wolf [3] proved that this conjecture is true for *completely bounded block multilinear forms* – a class of polynomials that captures a special kind of quantum query algorithms.

In this work, we show that Aaronson-Ambainis conjecture is true for a large fraction of random restrictions of f assuming $\mathsf{Var}[f]$ is not too low. More precisely, we have:

▶ **Theorem 5.** *There exist constants $C_1, C_2 > 0$ such that the following holds: let $f : \{\pm 1\}^n \to [0,1]$ be a degree d polynomial ($d \geq 2$) with $\mathsf{Var}[f] \geq 1/d$. Let ρ denote a random restriction with survival probability $\dfrac{\log(d)}{C_1 d}$. Then,*

$$\mathsf{Pr}\left[f_\rho \text{ has a coordinate with influence } \geq \frac{\mathsf{Var}[f]^2}{d^{C_2}}\right] \geq \frac{\mathsf{Var}[f]\log(d)}{50 C_1 d}.$$

We view this result as evidence that the Aaronson-Ambainis Conjecture is true, and hope it gives new insights into the conjecture.

We also observe that our result implies one of the two main unconditional results proven in a recent paper of Lovett and Zhang [15] about the existence of small sensitive blocks albeit with a slightly different set of parameters.

Let $f : \{\pm 1\}^n \to [0,1]$. An input $x \in \{\pm 1\}^n$ is said to be (r, ϵ) sensitive if there exists a y such that $d(x,y) \leq r$ and $|f(x) - f(y)| \geq \epsilon$. [15] proves the following:

▶ **Theorem 6** (Lovett and Zhang [15]). *If $f : \{\pm 1\}^n \to [0,1]$ has degree d, then at least $\Omega(\mathsf{Var}[f])$ fraction of the inputs are (r, ϵ) sensitive where $\epsilon = poly(\mathsf{Var}[f]/d), r = poly(d, 1/\epsilon, \log(n))$*

An immediate consequence of our result is that at least $\Omega((\mathsf{Var}[f]/d)^{O(1)})$ fraction of inputs are $(1, \epsilon)$ sensitive where $\epsilon = \mathrm{poly}(\mathsf{Var}[f]/d)$. Thus, while we lose a bit in the fraction of sensitive inputs, we gain by letting our block size be exactly 1 instead of $\mathrm{poly}(d, 1/\epsilon, \log(n))$.

It is interesting to remark that our techniques and those of Lovett and Zhang seem quite different. We further remark that [15] shows that if one can show *every* input has a small sensitive block, Aaronson-Ambainis conjecture will follow.

We believe that Theorem 5 opens up some possible lines of attack to the Aaronson-Ambainis conjecture:

- Assuming a supposed counterexample $f : \{\pm 1\}^n \to [0,1]$, modify it appropriately (e.g., by composing it with some appropriate gadget or applying a low noise operator) to get a function $\tilde{f} : \{\pm 1\}^{\tilde{n}} \to [0,1]$ such that most of its random restrictions remain a counterexample. Combined with our result, this will prove Aaronson-Ambainis conjecture. This approach is discussed in a bit more detail in the conclusion.
- If we can construct a decision tree that queries y (the input to the fixed part) in a few coordinates and outputs an influential coordinate of f_y (the restricted function) with high probability, Aaronson-Ambainis conjecture will follow from the work of Keller and Klein. Again, this approach is discussed in a bit more detail in the conclusion.

2 Organization

We introduce notations and necessary preliminaries in section 3. We give a high level overview of our proof in section 4. In section 5 we compile some lemmas that will be needed in our main proof. Our main results are proven in section 6. Our main result is proven in section 6.2. It says that most random restrictions of a bounded low-degree function can be approximated by a small junta. In section 6.3 we prove that Aaronson-Ambainis conjecture is true for a non-negligible fraction of random restrictions.

3 Notations and preliminaries

Query algorithms

1. A classical query algorithm A (or equivalently, a decision tree) for computing a function $f : \{\pm 1\}^n \to \mathbb{R}$ can access the input $x \in \{\pm 1\}^n$ by adaptively issuing queries to its bits. We assume internal computations have no cost. The depth of the query algorithm/decision tree is the maximum number of bit queries issued on an input. We say A ϵ-approximates f if $||f - A||_2^2 = \mathsf{E}_{x \in \{\pm 1\}^n}\left[(f(x) - A(x))^2\right] \leq \epsilon$.
   For a partial function $f : S(\subseteq \{\pm 1\}^n) \to \{0,1\}$, its classical query complexity $D(f)$ is the smallest d for which there exists a decision tree T of depth d such that $T(x) = f(x)$ for all $x \in S$.
2. A quantum query algorithm can access the input $x \in \{\pm 1\}^n$ via an oracle O_x. The register has n qubits and some ancilla qubits. The oracle O_x applies the following unitary operation on the first n qubits:

   $$O_x|s_1 s_2 \cdots s_n\rangle = (-1)^{\langle x,s\rangle}|s_1 s_2 \cdots s_n\rangle.$$

   The quantum query algorithm applies a sequence of unitary operators, where each operator is either O_x or an input-independent unitary operator U. In the end, it measures the first qubit and outputs the measurement result. The number of queries issued is the number of times O_x is applied.
   Notice that a quantum query algorithm Q naturally defines a function $P : \{\pm 1\}^n \to [0,1]$:

   $$P(x) = \Pr[Q \text{ outputs 1 on input } x].$$

   It is well-known that if Q makes q queries, then P is a degree $2q$ polynomial.

For a function $f : S(\subseteq \{\pm 1\}^n) \to \{0,1\}$, we define its quantum query complexity $Q(f)$ to be the smallest q for which there exists a quantum query algorithm Q making q queries such that for all $x \in S$,

$$\Pr[Q \text{ outputs 1 on input } x] \begin{cases} \geq 2/3 & \text{if } f(x) = 1 \\ \leq 1/3 & \text{if } f(x) = 0. \end{cases}$$

Analysis of boolean functions

In this section we recall some results from analysis of boolean functions. A good reference is O' Donnell's textbook [17].

1. Any function $f : \{\pm 1\}^n \to \mathbb{R}$ has a unique representation as $f(x) = \sum_{S \subseteq [n]} \hat{f}(S)\chi_S(x)$ where $\chi_S(x) = \prod_{i \in S} x_i$. The coefficients $\hat{f}(S)$ are the Fourier coefficients of f. The degree of f is $\max\{|S||\hat{f}(S) \neq 0\}$.
2. The variance of f is
$$\mathsf{Var}_{x \in \{\pm 1\}^n}[f(x)] = \sum_{S \neq \phi} \hat{f}(S)^2.$$
3. For a cooordinate i, the influence of the i'th coordinate is defined as
$$\begin{aligned} \mathsf{Inf}_i[f] &= \mathsf{E}_{x \in \{\pm 1\}^n}\left[\left(\frac{f(x) - f(x^{(i)})}{2}\right)^2\right] \\ &= \sum_{i \in S} \hat{f}(S)^2. \end{aligned}$$
The total influence of f is
$$\mathsf{Inf}[f] = \sum_{i \in [n]} \mathsf{Inf}_i[f] = \sum_S |S|\hat{f}(S)^2.$$
From the Fourier expansion it is clear that if $\deg(f) \leq d, \mathsf{Inf}[f] \leq d\mathsf{Var}[f]$.
4. Given two functions $f, g\{\pm 1\}^n \to \mathbb{R}$, we say g $\epsilon-$approximates f if $||f - g||_2^2 = \mathsf{E}_x\left[(f(x) - g(x))^2\right] \leq \epsilon$.
5. For a point $x \in \{\pm 1\}^n$ and a subset $S \subseteq [n]$ and $-1 \leq \rho \leq 1$, we define a distribution $N_{\rho,S}(x)$ on $\{\pm 1\}^n$ as follows:
   - The bits $y_1, y_2, \cdots, y_n$ are independent, and
$$\mathsf{Pr}[y_i = x_i] = \begin{cases} 1 & \text{if } i \notin S \\ (1+\rho)/2 & \text{if } i \in S. \end{cases}$$
When $S = [n]$, we abbreviate $N_{\rho,S}(x)$ by $N_\rho(x)$.
6. For $f : \{\pm 1\}^n \to \mathbb{R}$ and $-1 \leq \rho \leq 1$ define $T_\rho f : \{\pm 1\}^n \to \mathbb{R}$ by
$$T_\rho f(x) = \mathsf{E}_{z \leftarrow N_\rho(x)}[f(z)].$$
It is easy to see that the Fourier expansion of $T_\rho f$ is given by
$$T_\rho f(x) = \sum_{S \subseteq [n]} \rho^{|S|}\hat{f}(S)\chi_S(x).$$

7. A function $f : \{\pm 1\}^n \to \mathbb{R}$ is a *junta of arity l* or *l-junta* if there exists a subset $S \subseteq [n], |S| \leq l$ such that f only depends on the coordinates in S. We say f is a (ϵ, l) junta if it can be ϵ-approximated by a l-junta, i.e., there exists a l-junta g such that $||f - g||_2^2 \leq \epsilon$.
8. A restriction $\rho = (S, y)$ of $f : \{\pm 1\}^n \to \mathbb{R}$ is specified by a subset $S \subseteq [n]$ and an assignment $y \in \{\pm 1\}^{[n]\setminus S}$. Such a restriction naturally induces a function $f_\rho : \{\pm 1\}^S \to \mathbb{R}$. Sometimes we shall write f_y instead of f_ρ (note that S is determined by y since $y \in \{\pm 1\}^{[n]\setminus S}$).
   By a random restriction with survival probability p, we mean sampling $\rho = (S, y \in \{\pm 1\}^{[n]\setminus S})$ where each coordinate $i \in [n]$ is included in S with probability p independently, and each bit of y is independently set to a uniformly random bit.

▶ Remark 7. Throughout the paper, all growing parameters (e.g., the degree d) will be assumed to be larger than some sufficiently big constant. This is to make the expressions look neat, as we will be replacing terms like $(C_1)^k \text{poly}(k)$ by $(C_2)^k$ where $C_2 > C_1$.

4 Proof Overview

The main result in this paper is a structural result for bounded low-degree functions similar in spirit to Hastad's switching lemma [10]. Roughly speaking, we show that for come constant $C > 0$, the following holds: let $f : \{\pm 1\}^n \to [0, 1]$ be a polynomial of degree d, and let ρ denote a random restriction with survival probability $\frac{\log(d)}{Cd}$. Then,

$$\Pr\left[f_\rho \text{ is a } (O(d^{-C}), O(d^C)) \text{ junta}\right] \geq 1 - \frac{1}{d^{\Omega(1)}}.$$

Once this is established, we can prove Aaronson-Ambainis conjecture for random restrictions as follows: it is easy to see that

$$\mathsf{Pr}\left[\mathsf{Var}[f_\rho] \geq \frac{\mathsf{Var}[f]\log(d)}{2Cd}\right] \geq \frac{\mathsf{Var}[f]\log(d)}{2Cd}.$$

This probability will be significantly more than the failure probability of the main result $\left(\frac{1}{d^{\Omega(1)}}\right)$ (this is the only place where we need the lower bound on $\mathsf{Var}[f]$). So for a $\Omega(\mathsf{Var}[f]\log(d)/d)$ fraction of random restrictions, the variance of f_ρ is high *and* f_ρ can be well approximated by a junta with arity $\mathsf{poly}(d)$. This means one of the coordinates of the junta must have high influence. This concludes the proof.

Now we give a brief overview of how we prove the main result (Theorem 1). The starting point is the work by Dinur, Friedgut et al. [7] which states the following:

▶ **Theorem 8** (Dinur et al. [7]). *For any $f : \{\pm 1\}^n \to [0, 1]$, if*

$$\sum_{|S|>k} \hat{f}(S)^2 \leq \exp(-O(k^2 \log(k)/\epsilon)),$$

then f is a $(\epsilon, 2^{O(k)}/\epsilon^2)$ junta.

In other words, if the Fourier tail above a certain level k is bounded, then f can be well approximated by juntas of arity roughly $2^{O(k)}$.

We start with the observation that random restrictions have bounded Fourier tails (Lemma 15): if the function has degree d and we make a random restriction with survival probability $\frac{\log(d)}{Cd}$ where $C > 1$ is a large constant, using Chernoff bound we can show that with very high probability the Fourier weight above level $\log(d)$ will be low; around the order of $\exp(-\Omega(C\log(d)))$. If we can manage to bring the Fourier weight above $\log(d)$ small enough so that Theorem 8 applies, then we will get that f_ρ can be well approximated by a $\mathsf{poly}(d)$ junta. Unfortunately, if we try this, it turns out that we have to set the survival probability so low that on expectation the variance of f_ρ goes down significantly as well. In other words, while it is true that f_ρ can be well-approximated by juntas, it is for a trivial reason that its variance itself is very low. (And moreover, this is also true for functions $f : \{\pm 1\}^n \to \mathbb{R}$ that are merely L^2 bounded i.e, $\mathsf{E}[f^2] \leq 1$, so we would not be using any additional structure that arises from the fact that f is *pointwise* bounded.)

In order to make this approach work, we need to improve the tail bound $\exp(-O(k^2\log(k))/\epsilon)$ in Theorem 8. The problematic term is the quadratic $\exp(-O(k^2))$ in the exponential. If the dominant term were to the order of $\exp(-O(k))$ instead, the calculations would go through. Can we hope to increase the tail bound to $\exp(-O(k))$ while paying a cost by increasing the junta arity? Unfortunately, again, this is not possible: [7] constructs a function which shows that the tail bound is essentially tight upto the $\log(k)$ factor - their function has $||f^{>k}||_2^2 \approx \exp(-\Theta(k^2))$ but approximating it to even $1/3$ accuracy requires reading $\Omega(n)$ coordinates. Our key observation is that the function constructed by [7] has full degree whereas we are working with random restrictions of a low degree function, so in addition to the fact that the Fourier tail of f_ρ above level $\log(d)$ is very small, we also know that f_ρ has degree d. Can we hope to improve the tail bound in Theorem 8 if we have the additional restriction that the function is of degree d? Indeed, this turns out to be true. We prove the following result in section 6 (Theorem 28):

▶ **Theorem 9.** *There exists a constant C such that the following is true:*
If $f : \{\pm 1\}^n \to [0,1]$ has degree d and $\sum_{|S|>k} \hat{f}(S)^2 \leq \frac{\epsilon}{C^k d^C}$, f is a $(\epsilon, \epsilon^{-2}d^C C^k)$ junta.

Below we briefly discuss how we are able to improve the tail bound under the additional degree assumption. Dinur et al [7] prove their tail bound (Theorem 8) by showing the following result (we are omitting the exact quantitative parameters here for reading convenience).

▶ **Theorem 10.** *Let $h : \{\pm 1\}^n \to \mathbb{R}$ be a degree k function with $\mathsf{E}[h^2] \leq 1$ (but not necessarily pointwise bounded) which cannot be approximated by $2^{O(k)}$ juntas to accuracy μ. Then, for any function $g : \{\pm 1\}^n \to [0,1]$, $E[(h-g)^2] \geq \varepsilon$.*

Let's sketch how [7] proves Theorem 8 using Theorem 10. They assume f is not approximable by $2^{O(k)}$ juntas - their goal is then to show that the Fourier tail above level k, $||f^{\geq k}||_2^2$, is large. To do so, they use Theorem 10: they take h to be the truncated function $h = f^{\leq k}$ and g to be the original function $g = f$. This then lower bounds $\mathsf{E}[(f - f^{\leq k})^2]$ which is precisely the Fourier tail above weight k. Thus, the distance lower bound ε in Theorem 10 governs the Fourier tail lower bound in Theorem 8. Since in Theorem 28 we have the additional information that f is of degree d, for our purposes it will suffice to lower bound the distance of h from bounded degree d functions, not necessarily *all* bounded functions. In section 6.2 we prove a result of the following form (again, we are omitting the exact parameters for reading convenience; for the exact statement see Theorem 26).

▶ **Theorem 11** (Informal version of Theorem 26). *Let $h : \{\pm 1\}^n \to \mathbb{R}$ be a degree k function with $\mathsf{E}[h^2] \leq 1$ (but not necessarily pointwise bounded) which cannot be approximated by $2^{O(k)}$ juntas to accuracy μ. Then, for any degree d function $g : \{\pm 1\}^n \to [0,1]$, $E[(h-g)^2] \geq \tilde{\varepsilon}$.*

The parameter $\tilde{\varepsilon}$ in Theorem 11 is bigger than the corresponding ε parameter in Theorem 10 because we are only lower bounding the distance of h from bounded *low-degree* functions whereas Theorem 10 lower bounds the distance of h from *arbitrary* bounded functions. It turns out that the improvement in this parameter is sufficiently good for the calculations to go through.

In order to prove Theorem 11 we shall use the main idea of the proof of [7] along with a structural restriction for bounded low-degree functions proved first in Filmus et al. [8], and later used by Keller-Klein [13] and Lovett-Zhang [15]. Given any function $f : \{\pm 1\}^n \to \mathbb{R}$ and $x \in \{\pm 1\}^n$ define the *block sensitivity* of f at x to be

$$\mathrm{bs}(f,x) = \sup \left[\sum_{j \in [k]} \left| f(x) - f(x^{(B_j)}) \right| \right]$$

where the supremum ranges over all partitions $(B_1, B_2, \cdots, B_k)$ of the variables ($x^{(B_j)}$ denotes x with the coordinates in B_j flipped). Define the block sensitivity of f to be $\mathrm{bs}(f) = \sup_x \mathrm{bs}(f,x)$. We shall use the following fact about bounded low-degree functions:

▶ **Theorem 12.** *If $f : \{\pm 1\}^n \to [0,1]$ has degree d, $bs(f) \leq 6d^2$.*

We give a high-level overview of how we are able to improve upon the bound of ε using the fact that the block sensitivity of a bounded low degree function is small. At one point in their proof, [7] lower bounds the probability of a linear form of Rademacher random variables $l(x_1, x_2, \cdots, x_t) = a_1 x_1 + \cdots + a_t x_t$ exceeding a certain threshold times its standard deviation, i.e.,

$$\mathsf{Pr}\left[a_1 x_1 + \cdots + a_t x_t \geq \alpha \sqrt{a_1^2 + \cdots + a_t^2} \right].$$

(In the proof, $l(x_1, \cdots, x_t)$ is the linear part of a restriction of f: $\sum_{j \in U} \hat{f}_j(U) x_j$ for some $U \subseteq [n]$.)

For each such point x where this linear form is high, [7] shows that *many related points* x' must have $f(x') > 2$. (To be precise, these related points are obtained by applying an appropriate noise operator to x.) Using this they conclude that f must deviate from the interval $[0,1]$ too often and therefore cannot be approximated by any bounded function.

We follow the proof of [7] up until this point. Instead of directly lower bounding the probability that $a_1 x_1 + \cdots + a_t x_t$ is high, we partition the set of variables $[t]$ into L blocks $B_1, \cdots, B_L$ (L is an appropriately chosen parameter) such that each block gets roughly same total weight: for all $j \in [L]$,

$$\sum_{i \in B_j} a_i^2 \geq \frac{a_1^2 + \cdots + a_t^2}{2L}.$$

It will turn out that the a_i's are sufficiently small for such a partition to exist. For each block we lower bound the probability that the linear form restricted to this block is high:

$$\mathsf{Pr}\left[\sum_{j \in B_i} a_j x_j \geq \tilde{\alpha} \sqrt{\sum_{j \in B_i} a_j^2} \right].$$

Now, on a random assignment z, the linear form restricted to many of these blocks will be high. Take such a block B_i: $\sum_{j \in B_i} a_j z_j \geq \tilde{\alpha} \sqrt{\sum_{j \in B_i} a_j^2}$. For each such block we will be able to find a large number of related points z_i such that $|f(z) - f(z_i)|$ is large. (In our case, these related points are obtained by applying an appropriate noise operator restricted to B_i.) Crucially, these related points will differ from z only at B_i. Thus, we will find many points which differ from z at disjoint sets and whose f differ from z significantly. This will show that f cannot be too close to a bounded low degree function, because those functions have low block sensitivity.

Our advantage is that we need to set α' so that we can conclude $|f(z) - f(z')|$ is only somewhat larger than $\Omega(d^2/L)$ (as opposed to $\Omega(1)$ in [7]) - by setting L large enough this allows us to set a much smaller α and get rid of the quadratic exponential dependence.

5 Tools

In this section we compile some lemmas that we shall use in our proof.

A reverse Markov inequality

We will use the following simple inequality throughout the proof.

▶ **Lemma 13.** *Let X be a random variable such that $X \leq M$ with probability 1. Let $\mathsf{E}[X] = \mu > 0$. Then, $\mathsf{Pr}[X \geq \mu/2] \geq \frac{\mu}{2M}$.*

Proof. Assume $\mathsf{Pr}[X \geq \mu/2] < \frac{\mu}{2M}$. Then,

$$\mathsf{E}[X] \leq \mathsf{Pr}[X \geq \mu/2]M + \mathsf{Pr}[X \leq \mu/2]\frac{\mu}{2} < \mu,$$

contradiction. ◀

An anticoncentration inequality for linear forms of Rademacher random variables

▶ **Lemma 14.** *There exists a universal constant K such that the following holds: let $x_1, \cdots, x_n$ be independent Rademacher random variables and let $l(x_1, \cdots, x_n) = a_1 x_1 + \cdots + a_n x_n$. Let $\sigma = \sqrt{a_1^2 + \cdots + a_n^2}$. Suppose $|a_i| \leq \frac{\sigma}{Kt}$. Then,*

$$\Pr[l(x_1, \cdots, x_n) \geq t\sigma] \geq \exp(-Kt^2).$$

Proof. Equation 4.2 in [14]. ◀

Random restrictions have small tail

▶ **Lemma 15.** *Let $f : \{\pm 1\}^n \to \mathbb{R}$ have degree d, $C > 1$ be a sufficiently large constant, and let $\rho = (S, y \in \{\pm 1\}^{[n]\setminus S})$ be a random restriction with survival probability $\frac{\log(d)}{Cd}$. Let $k = \log(d)$. Then,*

$$\mathsf{E}\left[\sum_{|T|>k} \hat{f}_y(T)^2\right] \leq \exp(-C\log(d)/8)\mathsf{Var}[f].$$

Proof. First suppose $(S, y \in \{\pm 1\}^{[n]\setminus S})$ is a fixed restriction. Note that for $z \in \{\pm 1\}^S$,

$$f_y(z) = \sum_{U \subseteq [n]} \hat{f}(U)\chi_U(y, z)$$

so for $T \subseteq [S]$, $\hat{f}_y(T) = \sum_{U \subseteq S} \hat{f}(U \cup T)\chi_U(y)$. By Parseval's theorem, for a fixed S,

$$\mathsf{E}_y\left[\hat{f}_y(T)^2\right] = \sum_{U \subseteq [n]\setminus S} \hat{f}(U \cup T)^2.$$

Therefore, for a fixed S,

$$\mathsf{E}_y\left[\sum_{|T|>k} \hat{f}_y(T)^2\right] = \sum_{V \subseteq [n], |V \cap S|>k} \hat{f}(V)^2.$$

Randomizing over S again,

$$\mathsf{E}_{S,y}\left[\sum_{|T|>k} \hat{f}_y(T)^2\right] = \sum_{V \subseteq [n]} \Pr[|V \cap S| > k]\hat{f}(V)^2.$$

Since f has degree d, we only need to worry about the terms where $|V| \leq d$. Also, for $|V| \leq k$ the relevant probability is 0. Since each element is included in S with probability $\frac{\log(d)}{Cd}$, by Chernoff bound, for each V with $|V| \leq d$,

$$\Pr\left[|V \cap S| > k\right] \leq \exp((C-1)^2 k/4C) \leq \exp(-C\log(d)/8).$$

Thus we get that

$$\mathsf{E}_{S,y}\left[\sum_{|T|>k} \hat{f}_y(T)^2\right] \leq \exp(-C\log(d)/8) \sum_{T \neq \phi} \hat{f}(T)^2 = \exp(-C\log(d)/8)\mathsf{Var}[f]. \quad \blacktriangleleft$$

Random restrictions don't have low variance

▶ **Lemma 16.** *Let $f : \{\pm 1\}^n \to \mathbb{R}$ be any function and let $\rho = (S, y \in \{\pm 1\}^{[n]\setminus S})$ be a random restriction with survival probability p. Then,* $\mathsf{E}[\mathsf{Var}[f_\rho]] \geq p\mathsf{Var}[f]$.

Proof. Fix a restriction $(S, y \in \{\pm 1\}^{[n]\setminus S})$. For each $T \subseteq S$, $\hat{f}_y(T) = \sum_{U \subseteq [n]\setminus S} \hat{f}(T \cup U)\chi_U(y)$.

Thus, by Parseval's theorem, for a fixed S,

$$\mathsf{E}_y[\mathsf{Var}[f_y]] = \sum_{T : T \cap S \neq \phi} \hat{f}(T)^2.$$

Randomizing over S again,

$$\begin{aligned}
\mathsf{E}_{S,y}[\mathsf{Var}[f_y]] &= \sum_{T \neq \phi} \Pr[T \cap S \neq \phi]\hat{f}(T)^2 \\
&= \sum_{T \neq \phi} \left(1 - (1-p)^{|T|}\right)\hat{f}(T)^2 \\
&\geq \sum_{T \neq \phi} p\hat{f}(T)^2 \\
&= p\mathsf{Var}[f].
\end{aligned}$$

$\blacktriangleleft$

Random restrictions with appropriate survival probability put large Fourier mass on the linear level

▶ **Lemma 17** (implicit in [7]). *Let $f : \{\pm 1\}^n \to \mathbb{R}$, $J \subseteq [n]$ and k be such that*

$$\sum_{2^k \leq |T \cap J^c| < 2^{k+1}} \hat{f}(T)^2 \geq \mu.$$

Consider a random restriction $\rho = (S, y \in \{\pm 1\}^{[n]\setminus S})$ where each $j \in J$ is fixed and given a uniformly random assignment, and each $i \in J^c$ is kept alive with probability $p = 2^{-k}$. Then,

$$\mathsf{E}\left[\sum_{i \in S} \hat{f}_y(\{i\})^2\right] \geq \frac{\mu}{20}.$$

Proof. For a fixed $(S, y \in \{\pm 1\}^{[n]\setminus S})$ (note that $J \cap S = \phi$) and $j \in S$ we have

$$\hat{f}_y(\{j\}) = \sum_{T \subseteq [n], T \cap S = \{j\}} \hat{f}(T)\chi_{T\setminus\{j\}}(y).$$

By Parseval's theorem, for a fixed S,

$$\mathsf{E}_y\left[\hat{f}_y(\{j\})^2\right] = \sum_{T \subseteq [n], T \cap S = \{j\}} \hat{f}(T)^2.$$

Randomizing over S,

$$\begin{aligned}\mathsf{E}\left[\sum_{j \in S} \hat{f}_y(\{j\})^2\right] &= \sum_{T \subseteq [n]} \left[\sum_{j \in [n]} \Pr[T \cap S = \{j\}]\right] \hat{f}(T)^2 \\ &\geq \sum_{2^k \leq |T \cap J^c| < 2^{k+1}} |T| p (1-p)^{|T|-1} \hat{f}(T)^2.\end{aligned}$$

By standard inequalities, for $n \in [1/p, 2/p)$, $np(1-p)^{n-1} \geq 1/20$. It follows that

$$\mathsf{E}_y\left[\hat{f}_y(\{j\})^2\right] \geq \frac{\mu}{20}.$$ ◀

Some hypercontractive inequalities for low degree functions

These lemmas and their proofs can be found in [7].

▶ **Lemma 18** (Corollary 2.4 in [7]). *There exists a universal constant $W > 0$ such that the following holds:*

Let $f : \{\pm 1\}^n \to \mathbb{R}$ be a degree k function. Let $\mathsf{E}[f^2] = \sigma^2$. Then,

$$\mathsf{E}\left[f(z)^2 1_{f(z)^2 \leq W^k \sigma^2}\right] \geq \frac{1}{2}\mathsf{E}[f(z)^2].$$

▶ **Lemma 19** (Lemma 2.5 in [7]). *There exists a universal constant $B > 0$ such that the following holds:*

Let $f : \{\pm 1\}^n \to \mathbb{R}$ be a degree k function. Let $\rho \in [-1/2, 1/2]$ be a noise parameter, and $x_0 \in \{\pm 1\}^n$. Suppose $\mathsf{E}_{z \leftarrow N_\rho(x_0)}[f(z) - f(x_0)] = \mu \geq 0$. Then,

$$\Pr_{z \leftarrow N_\rho(x_0)}[f(z) - f(x_0) \geq \mu] \geq \frac{1}{B^k}.$$

The noise lemma

This is the main result of [7]. We use a slight variant. First we recall some known results from approximation theory.

▶ **Lemma 20.** *For any k, there exist constants $\rho_1, \rho_2, \cdots, \rho_{k+1} \in [-1/2, 1/2]$ with the following property: for any polynomial of degree k, $p(x) = a_0 + a_1 x + \cdots + a_k x^k$, there exists a $j \in [k+1]$ such that $p(\rho_j) \geq \frac{a_1}{2(k+1)}$.*

Proof. Page 112 in [20]. ◀

Now we state the lemma.

▶ **Lemma 21.** *There exists a universal constant $B > 0$ such that the following holds: Consider a degree k polynomial $f : \{\pm 1\}^n \to \mathbb{R}$. Let $S \subseteq [n]$ and $\ell(x) = \sum_{i \in S} \hat{f}(\{i\}) x_i$.*

Consider an input $x_0 \in \{\pm 1\}^n$ such that $\ell(x_0) \geq \gamma$. Sample a $z \leftarrow \{\pm 1\}^n$ by the following procedure:

1. *Sample $\rho \leftarrow \{\rho_1, \cdots, \rho_{k+1}\}$ uniformly at random.*
2. *Sample $z \leftarrow N_{\rho,S}(x_0)$.*

Then,

$$\Pr\left[f(z) - f(x_0) \geq \frac{\gamma}{2(k+1)}\right] \geq \frac{1}{(k+1)B^k}.$$

▶ Remark 22. Observe that z differs from x only in the coordinates of S. This will be crucial later on.

Proof. Take B to be the same universal constant as in Lemma 19. By replacing f with an appropriate restriction if necessary, we can assume $S = [n]$. Consider the polynomial $p(\rho) = T_\rho f(x_0) - f(x_0)$. From the Fourier expansion of noise operator, we see that

$$p(\rho) = \sum_{S \neq \phi} \rho^{|S|} \hat{f}(S).$$

This is a degree k polynomial in ρ with linear coefficient $l(x_0)$. By Lemma 20, there exists a $h \in [k+1]$ such that $p(\rho_h) \geq \gamma/(2k+2)$. By Lemma 19,

$$\Pr_{z \leftarrow N_\rho(x_0)}\left[f(z) - f(x_0) \geq \frac{\gamma}{2(k+1)} | \rho = \rho_h\right] \geq \frac{1}{B^k}.$$

We choose $\rho = \rho_h$ in step (1) with probability $1/(k+1)$, so

$$\Pr_{z \leftarrow N_\rho(x_0)}\left[f(z) - f(x_0) \geq \frac{\gamma}{2(k+1)}\right] \geq \frac{1}{(k+1)B^k}.$$ ◀

Partitioning a set of numbers in a balanced manner

We need an easy lemma about partitioning a set of weights none of which is too large into disjoint buckets where each bucket gets roughly the same total weight. We will later use this lemma on the set of small linear Fourier coefficients of a function.

▶ **Lemma 23.** *Let $a_1, a_2, \cdots, a_n$ be a set of non-negative real numbers and $1 \leq L \leq n$. Suppose $a_i \leq \frac{a_1 + a_2 + \cdots + a_n}{2L}$ for all $1 \leq i \leq n$. Then, there exists a partition $(B_1, B_2, \cdots, B_L)$ of $[n]$ such that for all $1 \leq j \leq L$,*

$$\sum_{i \in B_j} a_i \geq \frac{a_1 + \cdots + a_n}{2L}.$$

Proof. Start with an arbitrary partition $(B_1, B_2, \cdots, B_L)$. Then, refine it iteratively according to the following algorithm.

Refinement algorithm:

1. Locate a j such that the condition is violated for j, i.e.,

   $$\sum_{i \in B_j} a_i < \frac{a_1 + \cdots + a_n}{2L}.$$

   If no such j exists, terminate.
2. Locate a k such that

   $$\sum_{i \in B_k} a_i \geq \frac{a_1 + \cdots + a_n}{L}.$$

3. Take an arbitrary $l \in B_k$ such that $a_l \neq 0$ and place it in B_j;

   $$B_k \leftarrow B_k \setminus \{l\}$$

   $$B_j \leftarrow B_j \cup \{l\}$$

An appropriate k always exists in step (2) by an averaging argument. Since $a_l \leq \frac{a_1 + \cdots + a_n}{2L}$, the size of B_k does not go below $\frac{a_1 + \cdots + a_n}{2L}$ after step (3). It is easy to see this procedure must terminate. Formally, notice that the quantity

$$\sum_{j \in [L]} \min\left(\frac{a_1 + \cdots + a_n}{2L} - \sum_{i \in B_j} a_i, 0\right)$$

reduces by $\frac{1}{2L} \min\{a_i | a_i \neq 0\}$ at each step, so at some point of time it must be 0 at which point the algorithm terminates and returns a valid partition. ◀

Bounded low-degree functions have low block sensitivity

▶ **Lemma 24.** *Let $f : \{\pm 1\}^n \to [0, 1]$ be a polynomial of degree d. Then, $bs(f) \leq 6d^2$.*

Proof. Let $B_1, B_2, \cdots, B_k$ be any partition of $[n]$. Our goal is to show that for all $x \in \{\pm 1\}^n$,

$$\sum_{i=1}^{k} |f(x) - f(x^{(B_i)})| \leq 6d^2.$$

Let $g : \{\pm 1\}^k \to [0, 1]$ be the function obtained from f obtained by identifying the variables of B_i with a single variable y_i for $1 \leq i \leq k$. By Proposition 3.7 in [8], we have, for every

$y \in \{\pm 1\}^k$,

$$\sum_{i=1}^{k} |g(y) - g(y^{(i)})| \leq 6d^2.$$

This implies

$$\sum_{i=1}^{k} |f(x) - f(x^{(B_i)})| \leq 6d^2,$$

as desired. ◀

6 Main results

6.1 Improved tail bound for low degree functions

This section is the core technical part of our work. In Theorem 26 we show that if we have a function $f : \{\pm 1\}^n \to \mathbb{R}$ *(not necessarily bounded)* with $\mathsf{E}[f^2] \leq 1$ which cannot be approximated by juntas, then f cannot be well-approximated by bounded low-degree functions. In Theorem 28 we use Theorem 26 on the truncated function $f^{\leq k}$ to conclude that the tail bound appearing in [7] (Theorem 8) can be improved under the additional assumption that f has degree d.

▶ Remark 25. For a subset $J \subseteq [n]$, consider the junta $u : \{\pm 1\}^n \to \mathbb{R}$ which reads the coordinates of J and outputs the average over the unqueried coordinates. It is easy to see that $u(x) = \sum_{S \subseteq J} \hat{f}(S)\chi_S(x)$, so $||u - f||_2^2 = \sum_{S \not\subseteq J} \hat{f}(S)^2$. Thus, u approximates f if and only if $\sum_{S \not\subseteq J} \hat{f}(S)^2$ is small.

In fact, it is easy to see that there exists a function u depending only on coordinates of J such that $||f - u||_2^2 \leq \epsilon$ if and only if $\sum_{S \not\subseteq J} \hat{f}(S)^2 \leq \epsilon$. This immediately follows from the Fourier expansion of $f - u$.

▶ **Theorem 26.** *There exists a constant C such that the following holds:*
Let $f : \{\pm 1\}^n \to \mathbb{R}$ be a degree k function (not necessarily bounded) with $\mathsf{E}[f^2] \leq 1$. Let $J = \{j | \mathsf{Inf}_j[f] \geq \theta\}$ where $\theta = \frac{\mu^2}{C^k d^C}$. If $\sum_{S \not\subseteq J} \hat{f}(S)^2 \geq \mu$, then for any degree d function $g : \{\pm 1\}^n \to [0, 1]$, $\mathsf{E}[(f(x) - g(x))^2] \geq \delta = \frac{\mu}{C^k d^C}$.

▶ Remark 27. Notice here that although f is not pointwise bounded, g is.

Proof. Let W, B, K be the universal constants from Lemma 18, Lemma 21 and Lemma 14 respectively. We take C to be a constant sufficiently larger than B, K, W.

There exists a t such that

$$\sum_{2^t \leq |S \cap J^c| < 2^{t+1}} \hat{f}(S)^2 \geq \frac{\mu}{\log(k)}.$$

Let $\rho = (U, y \in \{\pm 1\}^{[n] \setminus U})$ $(U \subseteq J^c)$ be a random restriction where each $j \in J$ is fixed to a uniformly random assignment, and survival probability for each $j \notin J$ is 2^{-t}. By Lemma 17,

$$\mathsf{E}\left[\sum_{j \in U} \hat{f}_y(\{j\})^2\right] \geq \frac{\mu}{20 \log(k)}.$$

Fix a U such that

$$\mathsf{E}_{y\in\{\pm1\}^{[n]\setminus U}}\left[\sum_{j\in U}\hat{f}_y(\{j\})^2\right]\geq\frac{\mu}{20\log(k)}.$$

By Parseval's theorem we have for all $j\in U$,

$$\mathsf{E}\left[\hat{f}_y(\{j\})^2\right]=\sum_{S\cap U=\{j\}}\hat{f}(S)^2\leq\mathsf{Inf}_j[f].$$

For each $y\in\{\pm1\}^{[n]\setminus U}$ define $\mathsf{SMALL}_y=\{j|\hat{f}_y(\{j\})^2\leq W^k\mathsf{Inf}_j[f]\}$. Observe that for all y,

$$\sum_{j\in\mathsf{SMALL}_y}\hat{f}_y(\{j\})^2\leq W^k\mathsf{Inf}[f]\leq k\cdot W^k\leq(2W)^k.$$

For each $j\in U$ we have from Lemma 18

$$\mathsf{E}\left[\hat{f}_y(\{j\})^2\mathsf{1}_{\hat{f}_y(\{j\})^2\leq W^k\mathsf{Inf}_j[j]}\right]\geq\frac{1}{2}\mathsf{E}\left[\hat{f}_y(\{j\})^2\right].$$

Thus,

$$\mathsf{E}_{y\in\{\pm1\}^{[n]\setminus U}}\left[\sum_{j\in\mathsf{SMALL}_y}\hat{f}_y(\{j\})^2\right]\geq\frac{\mu}{40\log(k)},$$

so applying Lemma 13 [2]

$$\mathsf{Pr}_{y\in\{\pm1\}^{[n]\setminus U}}\left[\sum_{j\in\mathsf{SMALL}_y}\hat{f}_y(\{j\})^2\geq\frac{\mu}{80\log(k)}\right]\geq\frac{\mu}{80\log(k)(2W)^k}\geq\frac{\mu}{(3W)^k}.$$

Call $y\in\{\pm1\}^{[n]\setminus U}$ for which $\sum_{j\in\mathsf{SMALL}_y}\hat{f}_y(\{j\})^2\geq\frac{\mu}{80\log(k)}$ to be *good*. Let $\mathrm{GOOD}=\{y\in\{\pm1\}^{[n]\setminus U}|y\text{ is good}\}$. Let $L=\left\lceil\frac{(2B)^kd^8}{\mathsf{Var}[f]}\right\rceil$. For each good y, choose a partition $\mathsf{DIVIDE}(y)=(B_1,B_2,\cdots,B_L)$ of SMALL_y such that for all $1\leq i\leq L$,

$$\sum_{j\in B_i}\hat{f}_y(\{j\})^2\geq\frac{\mu}{160L\log(k)}.$$

(If there are multiple such partitions, choose any one of them and call it $\mathsf{DIVIDE}(y)$.)

Our choice of parameters ensures that for all $j\in\mathsf{SMALL}_y$, $\hat{f}_y(\{j\})^2\leq\frac{\mu}{160L\log(k)}$, so such a partition exists by Lemma 23. Let $\rho_1,\rho_2,\cdots,\rho_{k+1}$ be the constants from Lemma 20.

Suppose, for the sake of contradiction, there exists a degree d polynomial $g:\{\pm1\}^n\to[0,1]$ such that $\mathsf{E}[(f(x)-g(x))^2]\leq\delta$. Throughout the rest of the proof, for a string $s_1\in\{\pm1\}^{[n]\setminus U}$ and a string $s_2\in\{\pm1\}^U$, the pair (s_1,s_2) denotes the string $s\in\{\pm1\}^n$ which agrees with s_1 on $[n]\setminus U$ and with s_2 on U. Consider the following randomized procedure which returns a real number.

[2] See remark 7.

Procedure 1:
1. Sample a $y \in \mathsf{GOOD}$ uniformly at random.
2. Sample $\rho \leftarrow \{\rho_1, \rho_2, \cdots, \rho_{k+1}\}$ uniformly at random.
3. Sample $z \leftarrow \{\pm 1\}^U$ uniformly at random.
4. Let $\mathsf{DIVIDE}(y) = (B_1, B_2, \cdots, B_L)$. Sample $\tilde{z}^{(i)} \leftarrow N_{B_i,\rho}(z)$ for $1 \le i \le L$.
5. Return $\sum_{i=1}^{L} |f(y,z) - f(y,\tilde{z}^{(i)})|$.

We estimate the probability that procedure 1 returns a number $> 15d^2$ in two different ways. First, we obtain a lower bound from the defintion of GOOD. Then, we obtain an upper bound using the assumption that $\mathsf{E}[(f(x) - g(x))^2] \le \delta$ and the fact that $\mathsf{Pr}[y \text{ is good}]$ is somewhat large. These two bounds will contradict each other - and that will prove the theorem.

Lower bound. Fix a $y \in \mathsf{GOOD}$. Let $\mathsf{DIVIDE}(y) = (B_1, B_2, \cdots, B_L)$.

Let $w = \sqrt{\dfrac{\mu}{160L\log(k)}}$. For each $i \in [L]$ we have

$$\sqrt{\sum_{j \in B_i} \hat{f}_y(\{j\})^2} \ge w.$$

Choose α such that $\alpha w = \dfrac{100d^4(2B)^k}{L}$. Our choice of L ensures that $\alpha \le 1$. Moreover, our choice for influence threshold θ ensures that $|\hat{f}_y(\{j\})| \le \dfrac{w}{K\alpha}$ for all $j \in B_i$ where K is the universal constant from the Lemma 14.

Therefore, we can apply Lemma 14 to obtain that

$$\mathsf{Pr}_{z \in \{\pm 1\}^{B_i}} \left[\sum_{j \in B_i} z_j \hat{f}_y(\{j\}) \ge \frac{100d^4(2B)^k}{L} \right] \ge \exp(-K\alpha^2) \ge \frac{1}{K_1}.$$

Here $K_1 = \exp(K)$ is an absolute constant.

By Lemma 21 applied on the restriction $f_y : \{\pm 1\}^U \to \mathbb{R}$, as we sample $z \leftarrow \{\pm 1\}^U$ u.a.r, $\rho \leftarrow \{\rho_1, \cdots, \rho_{k+1}\}$ u.a.r, $\tilde{z}^{(i)} \leftarrow N_{\rho,B_i}(z)$, we have that

$$\mathsf{Pr}\left[|f(y,z) - f(y,\tilde{z}^{(i)})| \ge \frac{30d^3(2B)^k}{L} \right] \ge \frac{1}{K_1(k+1)B^k} \ge \frac{1}{(2B)^k}.$$

By linearity of expectation,

$$\mathsf{E}\left[\left| \left\{ i \in [L] \middle| |f(y,z) - f(y,\tilde{z}^{(i)})| \ge \frac{30d^3(2B)^k}{L} \right\} \right| \right] \ge \frac{L}{(2B)^k}.$$

Using Lemma 13,

$$\mathsf{Pr}\left[\left| \left\{ i \in [L] \middle| |f(y,z) - f(y,\tilde{z}^{(i)})| \ge \frac{30d^3(2B)^k}{L} \right\} \right| \ge \frac{L}{2 \times (2B)^k} \right] \ge \frac{1}{2 \times (2B)^k}.$$

Observe that

$$\left| \left\{ i \in [L] \middle| |f(y,z) - f(y,\tilde{z}^{(i)})| \ge \frac{30d^3(2B)^k}{L} \right\} \right| \ge \frac{L}{2 \times (2B)^k} \implies \sum_{i \in [L]} |f(y,z) - f(y,\tilde{z}^{(i)})| \ge 15d^3.$$

We conclude that for all $y \in \mathsf{GOOD}$, as $z, \tilde{z}^{(1)}, \cdots, \tilde{z}^{(L)}$ are sampled as in Procedure 1,

$$\mathsf{Pr}\left[\sum_{i\in[L]} |f(y,z) - f(y,\tilde{z}^{(i)})| \geq 15d^3\right] \geq \frac{1}{2\times(2B)^k}.$$

Thus, with probability at least $\frac{1}{2\times(2B)^k}$, procedure 1 returns a number greater than $15d^2$(as $15d^3 > 15d^2$).

Upper bound. Since $\mathsf{E}[(f(x)-g(x))^2] \leq \delta$ and $\mathsf{Pr}_{y\in\{\pm1\}^{[n]\setminus U}}[y \text{ is good}] \geq \mu/(3W)^k$, we have that

$$\mathsf{E}[(f(x)-g(x))^2 | x_{[n]\setminus U} \text{ is good}] \leq \frac{\delta}{\mu}(3W)^k.$$

Now consider a uniformly sampled $y \in \mathsf{GOOD}$. Observe that as we sample $z \leftarrow \{\pm1\}^U$ u.a.r, $\rho \leftarrow \{\rho_1, \cdots, \rho_{k+1}\}$ u.a.r and $\tilde{z}^{(i)} \leftarrow N_{\rho,B_i}(z)$, the marginal distribution of $\tilde{z}^{(i)}$ is uniform on $\{\pm1\}^U$. By Markov's inequality, we have

$$\mathsf{Pr}\left[(f(y,z)-g(y,z))^2 \geq \frac{1}{L^2}\right] \leq \frac{L^2\delta}{\mu}(3W)^k$$

and for all $i \in [L]$,

$$\mathsf{Pr}\left[(f(y,\tilde{z}^{(i)})-g(y,\tilde{z}^{(i)}))^2 \geq \frac{1}{L^2}\right] \leq \frac{L^2\delta}{\mu}(3W)^k.$$

By union bound, the probability that $(f(y,z)-g(y,z))^2 \geq \frac{1}{L^2}$ or for some i, $(f(y,\tilde{z}^{(i)}) - g(y,\tilde{z}^{(i)}))^2 \geq \frac{1}{L^2}$ is at most $(L+1)\frac{L^2\delta}{\mu}(3W)^k \leq \frac{2L^3\delta}{\mu}(3W)^k$. Our choice of δ ensures that this quantity is less than $< \frac{1}{2\times(2B)^k}$. Observe that if none of these bad events holds, since the block sensitivity of g is bounded above by $6d^2$ (Theorem 24), we have that

$$\sum_{i\in[L]} |g(y,z)-g(y,\tilde{z}^{(i)})| \leq 6d^2 \implies \sum_{i\in[L]} |f(y,z)-f(y,\tilde{z}^{(i)})| \leq 6d^2+1 < 15d^2.$$

Thus, we conclude

$$\mathsf{Pr}\left[\sum_{i\in[L]} |f(y,z)-f(y,\tilde{z}^{(i)})| > 15d^2\right] < \frac{2L^3\delta}{\mu}(3W)^k < \frac{1}{2\times(2B)^k}.$$

As promised, we get conflicting lower and upper bounds for the probability that procedure 1 returns a number $> 15d^2$. This is our desired contradiction. ◀

Now we show that we can improve the tail bound of Theorem 8 under the additional assumption that f has low degree. This follows straightforwardly from Theorem 26.

▶ **Theorem 28.** *There exists a universal constant $C > 0$ such that the following is true: Let $f : \{\pm1\}^n \to [0,1]$ be a degree d function. Let $\theta = \frac{\mathsf{Var}[f]^2}{d^C C^k}$ and $J = \{j | \mathsf{Inf}_j[f] \geq \theta\}$. If $\sum_{S\not\subseteq J} \hat{f}(S)^2 \geq \mu$, then $\sum_{|S|>k} \hat{f}(S)^2 \geq \frac{\mu}{d^C C^k}$.*

Proof. Assume $\sum_{|S|>k} \hat{f}(S)^2 < \mu/2$ (otherwise we are done). Let $\tilde{C}$ be the universal constant from Theorem 26.

The idea is to apply Theorem 26 to the truncated function

$$f^{\leq k}(x) = \sum_{|S|\leq k} \hat{f}(S)\chi_S(x).$$

Note that while $f^{\leq k}$ is not pointwise bounded, it satisfies $\mathsf{E}[(f^{\leq k})^2] \leq 1$ and $\mathsf{Inf}_j[f^{\leq k}] \leq \mathsf{Inf}_j[f]$ for all j (this is clear from the Fourier expressions). Let $H = \{j|\mathsf{Inf}_j[f^{\leq k}] \geq \theta\}$. We have $H \subseteq J$, so

$$\sum_{S \not\subseteq H} \hat{f^{\leq k}}(S)^2 \geq \sum_{S \not\subseteq J} \hat{f}(S)^2 - \frac{\mu}{2} \geq \frac{\mu}{2}.$$

Applying Theorem 26, we get that for any bounded degree d $g : \{\pm 1\}^n \to [0,1]$, $\mathsf{E}[(f(x) - g(x))^2] \geq \frac{\mu}{2d^{\tilde{C}}\tilde{C}^k}$. Taking g to be our original function f, we get the desired tail lower bound:

$$\mathsf{E}[(f - f^{\leq k})^2] \geq \frac{\mu}{2d^{\tilde{C}}\tilde{C}^k} \implies \sum_{|S|>k} \hat{f}(S)^2 > \frac{\mu}{2d^{\tilde{C}}\tilde{C}^k}.$$

Taking C to be a slightly larger constant than $\tilde{C}$, we get that

$$\sum_{|S|>k} \hat{f}(S)^2 \geq \frac{\mu}{d^C C^k}.$$ ◀

6.2 Random restrictions can be approximated by juntas

In this section we use the fact that random restrictions have bounded tails to show that they can be approximated by juntas. This is our main result.

▶ **Theorem 29 (Restatement of Theorem 1).** *For any constants $\tilde{C}_1, \tilde{C}_2 > 0$, there exist constants $\tilde{C}_3, \tilde{C}_4, \tilde{C}_5 > 0$ such that the following holds:*
Let $f : \{\pm 1\}^n \to [0,1]$ be a degree d polynomial and let ρ be a random restriction with survival probability $\frac{\log(d)}{\tilde{C}_3 d}$. With probability at least $1 - d^{-\tilde{C}_2}$, f_ρ is a $(d^{-\tilde{C}_1}\mathsf{Var}[f], \mathsf{Var}[f]^{-2}d^{\tilde{C}_4})$ junta. Moreover, if J denotes the set of coordinates on which the junta depends, for each $j \in J$ we have $\mathsf{Inf}_j[f] \geq \mathsf{Var}[f]^2 d^{-\tilde{C}_5}$.

Proof. We consider a random restriction with survival probability $\frac{\log(d)}{\tilde{C}_3 d}$.

By Lemma 15, the expected Fourier tail of f_ρ above level $\log(d)$ is at most $\exp(-\tilde{C}_3 \log(d)/8)\mathsf{Var}[f] = \frac{\mathsf{Var}[f]}{d^{\tilde{C}_3/8}}$. By Markov's inequality, with probability at least $1 - d^{\tilde{C}_3/16}$, the Fourier tail above $\log(d)$ is $\leq \frac{\mathsf{Var}[f]}{d^{\tilde{C}_3/16}}$. Let C be the constant from Theorem 28. Let $\mu = \frac{\mathsf{Var}[f]}{d^{\tilde{C}_3/16}} d^C C^{\log(d)} = \frac{\mathsf{Var}[f]}{d^{\tilde{C}_3/16}} d^{2C}$, $\theta = \frac{\mu^2}{d^C C^{\log(d)}} = \frac{\mu^2}{d^{2C}}$ and $J = \{j|\mathsf{Inf}_j[f_\rho] \geq \theta\}$. Let $u_\rho : \{\pm 1\}^n \to [0,1]$ be the function which reads the coordinates in J and outputs the average of f_ρ over the coordinates in J^c. Choose $\tilde{C}_3$ large enough so that $\mu \leq d^{-\tilde{C}_1}\mathsf{Var}[f]$. Applying Theorem 28, we see that u_ρ approximates f_ρ to accuracy $d^{-\tilde{C}_1}\mathsf{Var}[f]$. Using the fact that total influence is bounded by d, we see that u_ρ has arity $\leq \mathsf{Var}[f]^{-2}d^{C'\tilde{C}_3}$ for a universal constant C'. Taking $(\tilde{C}_4, \tilde{C}_5) = (C'\tilde{C}_3, \tilde{C}_3/32 - 2C)$, we are done. ◀

6.3 Aaronson-Ambainis Conjecture is true for random restrictions

▶ **Theorem 30 (Restatement of Theorem 5).** *There exist constants $C_1, C_2 > 0$ such that the following holds: let $f : \{\pm 1\}^n \to [0,1]$ be a degree d polynomial ($d \geq 2$) with $\mathsf{Var}[f] \geq 1/d$. Let ρ denote a random restriction with alive probability $\frac{\log(d)}{C_1 d}$. Then,*

$$\mathsf{Pr}\left[f_\rho \text{ has a coordinate with influence } \geq \frac{\mathsf{Var}[f]^2}{d^{C_2}}\right] \geq \frac{\mathsf{Var}[f]\log(d)}{50 C_1 d}.$$

Proof. Let M be a large constant. Apply Theorem 29 with $(\tilde{C}_1, \tilde{C}_2) = (M, M)$ to get constants $\tilde{C}_3, \tilde{C}_4, \tilde{C}_5$. Let ρ be a random restriction with survival probability $\frac{\log(d)}{\tilde{C}_3 d}$. By Lemma 16,

$$\mathsf{E}[\mathsf{Var}[f_\rho]] \geq \frac{\mathsf{Var}[f]\log(d)}{\tilde{C}_3 d}$$

so by Lemma 13,

$$\mathsf{Pr}\left[\mathsf{Var}[f_\rho] \geq \frac{\mathsf{Var}[f]\log(d)}{2\tilde{C}_3 d}\right] \geq \frac{\mathsf{Var}[f]\log(d)}{2\tilde{C}_3 d}.$$

Since $\mathsf{Var}[f] \geq 1/d$, $d^{-M} \leq \frac{\mathsf{Var}[f]\log(d)}{10\tilde{C}_3 d}$. By Theorem 29 and Remark 25, with probability at least $1 - d^{-M}$, there exists a $J_\rho \subseteq [n]$ such that every coordinate in J_ρ has influence $\geq \mathsf{Var}[f_\rho]^2 d^{-\tilde{C}_5}$ and

$$\sum_{S \not\subseteq J_\rho} \hat{f}_\rho(S)^2 \leq d^{-M}\mathsf{Var}[f].$$

So with probability at least $\frac{\mathsf{Var}[f]\log(d)}{2\tilde{C}_3 d} - d^{-M} \geq \frac{\mathsf{Var}[f]\log(d)}{4\tilde{C}_3 d}$, both these events (high variance of f_ρ and existence of J_ρ) hold and we have that

$$\sum_{S \subseteq J_\rho} \hat{f}_\rho(S)^2 \geq \mathsf{Var}[f_\rho] - d^{-M}\mathsf{Var}[f] \geq \frac{\mathsf{Var}[f]\log(d)}{4\tilde{C}_3 d}.$$

In particular, we have that $J_\rho \neq \phi$. Since for each $j \in J_\rho$ we have $\mathsf{Inf}_j[f_\rho] \geq \mathsf{Var}[f_\rho]^2 d^{-\tilde{C}_5}$, we are done by taking $(C_1, C_2) = (\tilde{C}_3, 2 + \tilde{C}_5)$. ◀

7 Conclusions and further directions

In this paper, we showed that if $f : \{\pm 1\}^n \to [0,1]$ is a degree d polynomial, a large fraction of its random restrictions have an influential coordinate.

It would be interesting to see if we can extend this to the full Aaronson-Ambainis conjecture. We describe a few potential approaches here.

- Given a degree d polynomial $f : \{\pm 1\}^n \to [0,1]$, we can lift it with a Boolean function $g : \{\pm 1\}^m \to \{\pm 1\}^n$ each of whose coordinates g_i is unbiased and given by a low degree function. Then, the lifted polynomial $f \odot g : \{\pm 1\}^m \to [0,1]$ will be a low degree polynomial. As long as the g_i's are pairwise independent, the variance of f will be preserved as well. Our result shows that a large fraction of random restrictions of

$f \odot g$ have an influential coordinate. Can we construct $g_1, g_2, \cdots, g_n$ appropriately such that this allows us to conclude f must have an influential coordinate as well? The g_i's should introduce correlations between the different input bits of f so that most random restrictions of $f \odot g^m$ *look the same* in some appropriate sense.

- Our result shows that for a non-negligible fraction of $y \in \{\pm 1\}^{[n]\setminus U}$, there exists a j such that $\mathsf{Inf}_j[f_y] \geq \mathrm{Var}[f]^2/d^{O(1)}$. Can we construct a decision tree of height $\mathrm{poly}(d)$ which, on input y, outputs one of the influential coordinates of f_y with high probability? If this is possible, this will imply Aaronson-Aambainis via the approach of Keller and Klein. We may additionally assume that for all j, $\mathrm{Inf}_j[f] = \mathrm{E}_y[\mathrm{Inf}_j[f_y]] \leq \mathrm{Var}[f]^2/d^{10000}$ (otherwise the result holds for f anyway). So basically the situation looks like this: for a non-negligible fraction of y, for some j, $\mathrm{Inf}_j[f_y]$ is significantly higher than its expectation ($\mathrm{Inf}_j[f]$) – and we want to probe y in a few bits to locate such a j.

References

1. Scott Aaronson and Andris Ambainis. Forrelation: A problem that optimally separates quantum from classical computing. *Electron. Colloquium Comput. Complex.*, TR14-155, 2014. URL: https://eccc.weizmann.ac.il/report/2014/155.
2. Scott Aaronson and Andris Ambainis. The need for structure in quantum speedups. *Theory Comput.*, 10:133–166, 2014. doi:10.4086/TOC.2014.V010A006.
3. Nikhil Bansal, Makrand Sinha, and Ronald de Wolf. Influence in completely bounded block-multilinear forms and classical simulation of quantum algorithms. In Shachar Lovett, editor, *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*, volume 234 of *LIPIcs*, pages 28:1–28:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.CCC.2022.28.
4. Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. In *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*, pages 352–361. IEEE Computer Society, 1998. doi:10.1109/SFCS.1998.743485.
5. Paul Beame, Russell Impagliazzo, Jan Krajícek, Toniann Pitassi, Pavel Pudlák, and Alan R. Woods. Exponential lower bounds for the pigeonhole principle. In S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis, editors, *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, pages 200–220. ACM, 1992. doi:10.1145/129712.129733.
6. Ethan Bernstein and Umesh V. Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, 1997. doi:10.1137/S0097539796300921.
7. Irit Dinur, Ehud Friedgut, Guy Kindler, and Ryan O'Donnell. On the fourier tails of bounded functions over the discrete cube. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pages 437–446, New York, NY, USA, 2006. Association for Computing Machinery. doi:10.1145/1132516.1132580.
8. Yuval Filmus and Hamed Hatami. Bounds on the sum of L1 influences. *CoRR*, abs/1404.3396, 2014. arXiv:1404.3396.
9. Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Math. Syst. Theory*, 17(1):13–27, 1984. doi:10.1007/BF01744431.
10. Johan Håstad. Almost optimal lower bounds for small depth circuits. In Juris Hartmanis, editor, *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*, pages 6–20. ACM, 1986. doi:10.1145/12130.12132.
11. Johan Håstad. On small-depth frege proofs for PHP. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 37–49. IEEE, 2023. doi:10.1109/FOCS57990.2023.00010.

12 Russell Impagliazzo, Raghu Meka, and David Zuckerman. Pseudorandomness from shrinkage. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 111–119. IEEE Computer Society, 2012. doi:10.1109/FOCS.2012.78.

13 Nathan Keller and Ohad Klein. Quantum speedups need structure, 2019. arXiv:1911.03748.

14 Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces*. Springer Berlin Heidelberg, 1991. doi:10.1007/978-3-642-20212-4.

15 Shachar Lovett and Jiapeng Zhang. Fractional Certificates for Bounded Functions. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 84:1–84:13, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2023.84.

16 Ashley Montanaro. Some applications of hypercontractive inequalities in quantum information theory. *Journal of Mathematical Physics*, 53(12), December 2012. doi:10.1063/1.4769269.

17 Ryan O'Donnell. Analysis of boolean functions, 2021. arXiv:2105.10386.

18 Ryan O'Donnell, Michael E. Saks, Oded Schramm, and Rocco A. Servedio. Every decision tree has an influential variable. *CoRR*, abs/cs/0508071, 2005. arXiv:cs/0508071.

19 Ryan O'Donnell and Yu Zhao. Polynomial bounds for decoupling, with applications. *CoRR*, abs/1512.01603, 2015. arXiv:1512.01603.

20 Theodore J. Rivlin. *Chebyshev polynomials – From approximation theory to algebra and number theory: 2nd edition*. John Wiley & Sons Limited, 1990.

21 Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. An average-case depth hierarchy theorem for boolean circuits. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1030–1048. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.67.

Finite Matrix Multiplication Algorithms from Infinite Groups

Jonah Blasiak
Department of Mathematics, Drexel University, Philadelphia, PA, USA

Henry Cohn
Microsoft Research New England, One Memorial Drive, Cambridge, MA, USA

Joshua A. Grochow
Departments of Computer Science and Mathematics, University of Colorado Boulder, CO, USA

Kevin Pratt
Department of Computer Science, Courant Institute of Mathematical Sciences, New York, NY, USA

Chris Umans
Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA

Abstract

The Cohn–Umans (FOCS '03) group-theoretic framework for matrix multiplication produces fast matrix multiplication algorithms from three subsets of a finite group G satisfying a simple combinatorial condition (the Triple Product Property). The complexity of such an algorithm then depends on the representation theory of G. In this paper we extend the group-theoretic framework to the setting of *infinite* groups. In particular, this allows us to obtain constructions in Lie groups, with favorable parameters, that are provably impossible in finite groups of Lie type (Blasiak, Cohn, Grochow, Pratt, and Umans, ITCS '23). Previously the Lie group setting was investigated purely as an analogue of the finite group case; a key contribution in this paper is a fully developed framework for obtaining bona fide matrix multiplication algorithms directly from Lie group constructions.

As part of this framework, we introduce "separating functions" as a necessary new design component, and show that when the underlying group is $G = \mathrm{GL}_n$, these functions are polynomials with their degree being the key parameter. In particular, we show that a construction with "half-dimensional" subgroups and optimal degree would imply $\omega = 2$. We then build up machinery that reduces the problem of constructing optimal-degree separating polynomials to the problem of constructing a *single* polynomial (and a corresponding set of group elements) in a ring of invariant polynomials determined by two out of the three subgroups that satisfy the Triple Product Property. This machinery combines border rank with the Lie algebras associated with the Lie subgroups in a critical way.

We give several constructions illustrating the main components of the new framework, culminating in a construction in a special unitary group that achieves separating polynomials of optimal degree, meeting one of the key challenges. The subgroups in this construction have dimension *approaching* half the ambient dimension, but (just barely) too slowly. We argue that features of the classical Lie groups make it unlikely that constructions in these particular groups could produce nontrivial bounds on ω unless they prove $\omega = 2$. One way to get $\omega = 2$ via our new framework would be to lift our existing construction from the special unitary group to GL_n, and improve the dimension of the subgroups from $\frac{\dim G}{2} - \Theta(n)$ to $\frac{\dim G}{2} - o(n)$.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Fast matrix multiplication, representation theory, infinite groups

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.18

Related Version *Full Version*: https://arXiv.org/abs/2410.14905 [10]



© Jonah Blasiak, Henry Cohn, Joshua A. Grochow, Kevin Pratt, and Chris Umans;
licensed under Creative Commons License CC-BY 4.0



Funding *Jonah Blasiak*: Supported by NSF grant DMS-2154282.
Joshua A. Grochow: Supported by NSF CAREER award CCF-2047756.
Kevin Pratt: Supported by Subhash Khot's Simons Investigator Award.
Chris Umans: Supported by a Simons Foundation Investigator Award.

Acknowledgements We are grateful to Peter Bürgisser and Emma Church for useful discussions, and we thank the American Institute for Mathematics for hosting a SQuaRE, during which initial parts of this work were developed.

1 Introduction

Matrix multiplication is a fundamental algebraic operation with myriad algorithmic applications, and as such, determining its complexity is a central question in computational complexity. Since Strassen's 1969 discovery [26] that one could beat the straightforward $O(n^3)$ method with one that used only $O(n^{\log_2 7}) = O(n^{2.81\cdots})$ arithmetic operations, there has been a long line of work improving upper bounds on the complexity of matrix multiplication. It is standard to define the exponent ω of matrix multiplication as the smallest number such that $n \times n$ matrices can be multiplied using $n^{\omega+o(1)}$ arithmetic operations, and a somewhat surprising folklore conjecture is that $\omega = 2$, which would mean matrices can be multiplied asymptotically almost as quickly as they can be added. The current best bound is that $\omega \leq 2.371339$ [29], and proving or disproving that $\omega = 2$ remains a longstanding open question.

In [14], an approach towards this problem was proposed based on embedding matrix multiplication into the multiplication operation in the group algebra of a finite group. Group algebra multiplication then reduces to the multiplication of block-diagonal matrices, where the sizes of these blocks are determined by the (typically well-understood) representation theory of the group. Ultimately, one hopes to reduce a single matrix multiplication to the multiplication of (many) smaller matrices. Within this very general framework, certain families of groups have subsequently been shown to have structural properties that prevent a reduction that would yield $\omega = 2$ using this framework [7, 8, 9]. Other groups remain potentially viable, and the overall approach remains one of the two main lines of research towards improving upper bounds on ω, the other being the traditional "direct" tensor methods (e. g., [24, 27, 15, 28, 16, 21, 2, 18]), which also seem to be running up against several barrier results [3, 12, 1, 11].

In more detail, the reduction to group algebra multiplication is possible when one identifies a finite group G and sets $X, Y, Z \subseteq G$ satisfying the *triple product property* (TPP): for any $x, x' \in X$, $y, y' \in Y$, and $z, z' \in Z$,

$$xx'^{-1}yy'^{-1}zz'^{-1} = 1_G \iff x = x', y = y', z = z'.$$

If $X, Y, Z \subseteq G$ satisfy the TPP, then we can multiply two complex matrices A and B, of sizes $|X| \times |Y|$ and $|Y| \times |Z|$, resp., as follows. Index A with $X \times Y$, with $A[x, y]$ denoting the x, y entry of A, and index B with $Y \times Z$. Then we define the elements

$$\overline{A} = \sum_{x,y} A[x, y](xy^{-1}) \quad \text{and} \quad \overline{B} = \sum_{y,z} B[y, z](yz^{-1})$$

of the group ring $\mathbb{C}[G]$ and observe that the TPP implies that

$$\overline{A} \cdot \overline{B} = \sum_{x,z} (AB)[x, z](xz^{-1}) + E, \tag{1.1}$$

where $E \in \mathbb{C}[G]$ is supported on $XY^{-1}YZ^{-1} \setminus XZ^{-1}$. It is a standard fact of representation theory that, as algebras, $\mathbb{C}[G] \cong \bigoplus_i M_{d_i}(\mathbb{C})$, where M_{d_i} denotes the ring of $d_i \times d_i$ matrices, the sum is over the irreducible representations of G, and the d_i are the dimensions of those representations. This leads to the inequality:

▶ **Theorem 1.1** ([14, Theorem 4.1]). *If X, Y, Z satisfy the TPP in a finite group G, then*

$$(|X||Y||Z|)^{\omega/3} \leq \sum_i d_i^\omega,$$

where d_i are the dimensions of the irreducible representations of G.

The upper bound on ω from Theorem 1.1 depends on a trade-off between the size of the matrix multiplication that can be embedded into $\mathbb{C}[G]$ (reflected in $|X|, |Y|, |Z|$) and the dimensions of the irreducible representations of G. In abelian groups, the latter are optimal: all d_i are 1. However, already in [14] it was observed that abelian groups cannot do better than the trivial construction $X = G$ and $Y = Z = \{1\}$, and thus cannot yield any bound better than the trivial bound $\omega \leq 3$. It was shown in [13] that non-abelian groups can achieve highly nontrivial bounds on ω (including many of the state-of-the-art bounds over the last decade), but obtaining such bounds requires a careful interplay between the size of the construction and the representation dimensions. Several families of groups have been shown not to admit constructions for which Theorem 1.1 would give $\omega = 2$, although many families of groups remain possibilities.

Because of the difficulty of finding such constructions, it is useful to look for other potential sources of examples of groups and group-like objects that could yield such constructions. Cohn and Umans [14] gave a construction of a TPP triple in the infinite group $\mathrm{GL}_n(\mathbb{R})$, despite having, at the time, no way of getting a (finite) matrix multiplication algorithm from such a construction.

One natural approach (that ends up not working) is to take a construction in a Lie group and try to transfer it to a finite group of Lie type, such as taking a construction in $\mathrm{GL}_n(\mathbb{R})$ and attempting an analogous construction in $\mathrm{GL}_n(\mathbb{F}_q)$. Indeed, a construction in $\mathrm{GL}_n(\mathbb{R})$ was given in [14] using the lower unitriangular, orthogonal, and upper unitriangular subgroups; this inspired a nontrivial TPP in the finite group $\mathrm{SL}_2(\mathbb{F}_q)$, where the orthogonal group is replaced by matrices of the form $\begin{pmatrix} 1+a & a \\ -a & 1-a \end{pmatrix}$. However, in that example, we have $|X| = |Y| = |Z| = q$, but $\mathrm{SL}_2(\mathbb{F}_q)$ has irreducible representations of dimension $q+1$, and reducing a $q \times q$ matrix multiplication to a $(q+1) \times (q+1)$ matrix multiplication doesn't give any bound on ω. More generally, in [9] it was shown that one cannot achieve $\omega = 2$ via Theorem 1.1 in any finite groups of Lie type, ruling out any such construction in $\mathrm{GL}_n(\mathbb{F}_q)$ or similar groups (though possibilities remain open to use related groups such as direct products of finite groups of Lie type). Thus, the Lie-type constructions remained only an analogy.

1.1 First main contribution: algorithms from infinite groups

In this paper, one of our main innovations is to extend the group-theoretic framework [14] to allow finite matrix multiplication algorithms from constructions in arbitrary – even infinite – groups. To achieve this, rather than using the entire group algebra $\mathbb{C}[G]$, which is infinite-dimensional when G is infinite, we focus only on sets of functions $G \to \mathbb{C}$ that (1) are linearly computable from a finite set of finite-dimensional representations of G (even when G is infinite) and (2) "separate" the elements in the group algebra that are in the linear span of $XY^{-1}YZ^{-1}$ (the support of (1.1)), in a sense made precise below (Definition 2.1). Our first main theorem in this generalized framework is then:

▶ **Theorem A** (Theorem 2.2). *Let G be a group (not necessarily finite), with finite subsets X, Y, Z satisfying the TPP. If R_{sep} is a set of finite-dimensional complex representations of G whose matrix entries separate $XY^{-1}YZ^{-1}$ (see Definition 2.1), then*

$$(|X|\,|Y|\,|Z|)^{\omega/3} \leq \sum_{\rho \in R_{\text{sep}}} (\dim \rho)^{\omega}.$$

It is even conceivable that this result could be used to improve the bounds from known constructions of TPPs in finite groups, by using only a subset of the group's irreducible representations rather than all of them.

But perhaps the main payoff of Theorem A is that it allows, for the first time, the derivation of matrix multiplication algorithms from infinite groups. This opens a huge variety of potential constructions to explore, even beyond those in Lie groups that will be the focus of the rest of the paper.

1.2 Second main contribution: quantitative targets for proving $\omega = 2$ in classical Lie groups

Another main contribution in this paper is to develop a series of tools and techniques, and to identify key targets to aim for, for getting good constructions using Theorem A in Lie groups such as $\mathrm{GL}_n(\mathbb{R})$, $\mathrm{GL}_n(\mathbb{C})$, or the unitary group U_n. Lie groups are defined as groups that are also smooth manifolds, and where the group multiplication and inverse are continuous in terms of the manifold topology. In fact, all of our constructions in this paper will be in one of these three groups, though much of the machinery we develop works for general matrix Lie groups, and it would be interesting to explore constructions in other Lie groups such as symplectic groups, orthogonal groups, exceptional simple Lie groups, or nilpotent or solvable Lie groups.

Before coming to the constructions, we highlight how the framework of Theorem A allows us to take the analogy from [14, 9], and turn it into a formal implication whose conclusion is a bound on ω. To briefly recall the analogy: elementary arguments show that any TPP triple in a finite group must satisfy $|X|\,|Y|\,|Z| \leq |G|^{3/2}$, called the *packing bound* because of the nature of the proof. Because $\sum d_i^2 = |G|$ in finite groups, it follows that any sequence of constructions that achieves $\omega = 2$ via Theorem 1.1 (not our new Theorem A) must asymptotically *meet the packing bound*, in the sense that $|X|\,|Y|\,|Z| \geq |G|^{3/2-o(1)}$. The analogy studied in [14, 9] is to think of Lie subgroups of dimension d as "roughly corresponding" to finite subsets of size q^d. Under this analogy, if X_n, Y_n, Z_n are families of Lie subgroups of Lie groups G_n that satisfy the TPP, then meeting the packing bound is analogous to the condition

$$\dim X_n + \dim Y_n + \dim Z_n \geq (3/2 - o_n(1)) \dim G_n. \tag{1.2}$$

A simple construction in the original Cohn–Umans paper [14, Theorem 6.1] shows this is indeed possible (with additional such constructions developed in [9]), and this construction forms the basis for a running example we will use throughout this paper to illustrate the development of our new framework.

Running example in $\mathbf{GL}_n(\mathbb{R})$

▶ **Theorem 1.2** ([14, Theorem 6.1]). *Let $G = \mathrm{GL}_n(\mathbb{R})$ and let X be the subgroup of lower unitriangular matrices, $Y = \mathrm{O}_n(\mathbb{R})$ (the orthogonal matrices), and Z be the subgroup of upper unitriangular matrices. Then the triple X, Y, Z satisfies the TPP.*

The dimension of G is n^2, and the dimension of each of the three subgroups is $n^2/2 - n/2$, so this construction meets the packing bound in the sense of (1.2) [9].

But there still remains the issue of how to use such a construction, even in concert with Theorem A, to get upper bounds on ω. And for this we require a deeper dive into representation theory, which will lead us to the key quantitative goals for these constructions.

In the case of the groups $\mathrm{GL}_n(\mathbb{R})$, $\mathrm{GL}_n(\mathbb{C})$, and U_n, rather than focusing on choosing arbitrary collections of representations to play the role of R_{sep} in Theorem A, we can exploit a relationship between the representations of these groups and the set of all degree-d polynomials. Namely, the irreducible representations of these groups are indexed by integer partitions into at most n parts, and the matrix entries of the representations corresponding to partitions of d, taken all together, span precisely the set of all degree-d polynomial functions on the group. Careful quantitative estimates then lead us to key targets for the degree of functions that separate out the elements of $XY^{-1}YZ^{-1}$ compared to the size of the finite TPP construction:

▶ **Theorem B** (Corollary 2.8, summarized). *If $X_{q,n}, Y_{q,n}, Z_{q,n} \subseteq \mathrm{GL}_n(\mathbb{C})$ (or U_n) satisfy the TPP and have sizes at least $q^{n^2/2-o_n(n)}$, and there are separating polynomials for $(X_{q,n}, Y_{q,n}, Z_{q,n})$ of degree at most $q^{1+o_q(1)}$, then Theorem A implies $\omega = 2$.*

Note how our new framework (Theorem A) has allowed us to take the analogy above whereby d-dimensional subgroups correspond to finite sets of size q^d, and turn it into a theorem that actually implies a bound on ω out of any such construction, rather than merely being an analogy. However, in this setup, as $\dim G = n^2$, we see that the bound we need on the size of the TPP triple is not $|X||Y||Z| \geq q^{(3/2-o_n(1))\dim G}$ as suggested in the previous Lie analogy of the packing bound [14, 9], but is slightly tighter, of the form $q^{(3/2-o_n(1/n))\dim G}$, and the example in Theorem 1.2 above falls short of this latter bound.

Our challenge is thus to find TPP constructions in GL_n or U_n that meet the bounds of Theorem B: three subsets in GL_n or U_n satisfying the TPP, of size at least $q^{n^2/2-o_n(n)}$, and admitting separating polynomials of degree at most $q^{1+o_q(1)}$.

1.3 Third main contribution: optimal degree using border rank, Lie algebras, and invariant theory

Our third main contribution is to show how to very nearly meet the conditions of Theorem B, using border rank, Lie algebras, and invariant theory. We also believe these techniques will have further uses. Our main theorem coming out of these constructions is:

▶ **Theorem C** (Summary of Theorem 3.1). *For any n and q, there are three subsets $X_q, Y_q, Z_q \subseteq \mathrm{U}_n$, all of size at least $q^{n^2/4-n/4}$, which satisfy the TPP and admit border-separating polynomials of degree $O(q)$.*

Note that this falls short of the conditions needed for Theorem B in only two ways: we get a construction of size $q^{(1/2)\dim G-\Theta(n)}$ where $G = \mathrm{U}_n$, whereas Theorem B would require both that the construction be in GL_n, and that it approach half the dimension just slightly faster, with sets of size $q^{(1/2)\dim \mathrm{GL}_n - o_n(n)}$, rather than our current $q^{(1/2)\dim \mathrm{U}_n-\Theta(n)}$. In addition to the sizes being very nearly right, the degree bound *does* satisfy the degree bound required by Theorem B (even slightly better than is needed: we get $O(q)$ whereas Theorem B only needs degree at most $q^{1+o_q(q)}$).

While in principle all one needs here is a sequence of finite sets X_q, Y_q, Z_q for infinitely many q, an appealing way to get such a sequence is to find Lie subgroups or submanifolds $X, Y, Z \subseteq \mathrm{GL}_n(\mathbb{C})$ – as in Theorem 1.2 – and then let X_q be some nicely constructed finite subset of X, etc. For example, when X is the subgroup of lower unitriangular matrices, we can take X_q to be the the lower unitriangular matrices with entries in $[0, q] \cap \mathbb{Z}$. And indeed, this is how the construction of Theorem C proceeds.

To get our constructions, we will combine three additional ingredients on top of Theorems A and B: Lie algebras, border rank, and invariant theory. Here we give a brief overview of these ingredients and how they mix together.

Lie algebras. In Sophus Lie's original development of Lie groups in the late 1800s, he realized that many questions about these continuous groups can be reduced to simpler questions of linear algebra, by focusing on the corresponding Lie algebras, which are, in particular, vector spaces, rather than more complicated manifolds. The Lie algebra $\mathrm{Lie}(G)$ associated to a Lie group G is "just" the tangent space to G (remember G is also a manifold) at the identity element. The Lie algebra is then a vector space. While the group structure of G induces an algebraic structure on $\mathrm{Lie}(G)$, in this paper we will have no need of its algebraic structure. We will only need a few basic facts (see, e. g., [22, 4, 19]):

- The Lie algebra of $\mathrm{GL}_n(\mathbb{C})$ is $M_n(\mathbb{C})$, the space of all $n \times n$ complex matrices.
- The Lie algebra of the orthogonal group $O_n(\mathbb{R})$ consists precisely of all skew-symmetric real matrices.
- The Lie algebra of the unitary group consists of the skew-Hermitian matrices.
- If G is a matrix Lie group – a Lie group that is a subgroup and submanifold of $\mathrm{GL}_n(\mathbb{C})$ – then its Lie algebra $\mathrm{Lie}(G)$ is a linear subspace of matrices. And if $A \in \mathrm{Lie}(G)$, then for all sufficiently small $\varepsilon > 0$, $\exp(\varepsilon A)$ (using the ordinary power series for the matrix exponential) is in G.

As with many problems on Lie groups, we would like to take advantage of the simpler, linear-algebraic nature of Lie algebras in our constructions.

Border rank. It turns out that the key tool for using Lie algebras in our setting is the concept of *border rank*. Although known to Terracini 100 years ago, border rank was rediscovered in the context of matrix multiplication by Bini *et al.* [5, 6]. Bini had developed computer code to search for algebraic algorithms for matrix multiplication, and some of the coefficients in his numerical calculations kept going off to infinity. At first he thought this was an error in his code, but in fact it reflects the fundamental phenomenon of border rank: for each fixed size n_0, it is possible that there is a sequence of algorithms, none of which correctly multiply $n_0 \times n_0$ matrices, but which, *in the limit*, in fact do so. If the algorithms in the sequence use only r non-constant multiplications, we say that matrix multiplication has border rank at most r. The border rank is always at most the ordinary rank, and it turns out that the exponent of matrix multiplication is the same whether measured with ordinary rank or border rank. Border rank has played an important role in essentially all newly developed matrix multiplication algorithm since then.

A bit more formally, a "border algorithm" for matrix multiplication can be viewed as a single bilinear algorithm that has coefficients that are Laurent series in ε – that is, power series that allow finitely many negative powers of ε as well – and such that it computes matrix multiplication in the limit as $\varepsilon \to 0$, that is, it computes a function of the form $(A, B) \mapsto AB + O(\varepsilon)$. (Note that, despite the algorithm itself being allowed to contain $1/\varepsilon$ in its intermediate operations, corresponding to Bini's coefficients that were going off to infinity, the function computed at the end should have such negative powers of ε cancel.)

Combining Lie algebras and border rank. It turns out that combining border rank with Lie algebras is very natural; here we exhibit just two advantages to doing so. If $X, Y, Z \subseteq \mathrm{GL}_n(\mathbb{C})$ are Lie subgroups that satisfy the TPP (as in Theorem 1.2), then we can take advantage of the simple linear-algebraic nature of their Lie algebras to help construct finite subsets of

X, Y, Z. An example we will return to is that if $Y = O_n(\mathbb{R})$ is the orthogonal group, it is a bit tricky to choose $q^{\dim O_n}$ many elements of Y in a principled way directly. But since $\mathrm{Lie}(Y)$ consists of all the skew-symmetric matrices, we can get a finite subset of Y by simply considering

$$Y'_\varepsilon = \{\exp(\varepsilon A) : A \text{ skew-symmetric with all } A_{ij} \in [0, q] \cap \mathbb{Z}\}$$

for $\varepsilon > 0$ sufficiently small.

A second advantage can be gotten by not choosing ε as above to be a fixed but small value, but rather allowing to the ε used in the expression $\exp(\varepsilon A)$ to be the same as the parameter ε used in the definition of border rank. In this setting, instead of finding a set of functions that exactly separates $XY^{-1}YZ^{-1}$ according to Definition 2.1, we can find functions that do so only up to $O(\varepsilon)$ (Definition 2.5). This both gives us more freedom in the construction, and combines very naturally with the Lie algebraic construction suggested above. Namely, when all the elements of X', Y', Z' are of the form $\exp(\varepsilon A)$ for various A in their Lie algebras, we see that an expression of the form $xy^{-1}y'z^{-1}$ becomes

$$\exp(\varepsilon A)\exp(-\varepsilon B)\exp(\varepsilon B')\exp(-\varepsilon C) = I + \varepsilon(A - B + B' - C) + O(\varepsilon^2).$$

Our border-separating polynomials can then directly access $A - B + B' - C$ by subtracting off I and dividing by ε; this leaves additional $O(\varepsilon)$ terms, but in the border setting that is still allowed. Thus combining Lie algebras and border rank lets us shift the problem from finding separating polynomials in the entries of a product of matrices and their inverses to finding (border-)separating polynomials in the entries of the simpler linear combination $A - B + B' - C$.

However, the linear combination $A - B + B' - C$ still "mixes" entries from the three subalgebras, and this causes some difficulty in the the task of designing (border-)separating polynomials. Our final ingredient is to use invariant theory to simplify this task even further.

1.4 Fourth main contribution: leveraging invariant polynomials

If we restrict our attention to polynomials $p(M)$ that are invariant under left multiplication by X and right multiplication by Z^{-1}, that is, $p(xMz^{-1}) = p(M)$ for all $x \in X, z \in Z$, we can get direct access to the $B' - B$ term above. Namely, if p is such an invariant polynomial, then

$$\begin{aligned} p(\exp(\varepsilon A)\exp(-\varepsilon B)\exp(\varepsilon B')\exp(-\varepsilon C)) &= p(\exp(-\varepsilon B)\exp(\varepsilon B')) \\ &= p(I + \varepsilon(-B + B') + O(\varepsilon^2)), \end{aligned}$$

where the first equality occurs because $\exp(\varepsilon A)$ is in X and $\exp(\varepsilon C)$ is in Z, and p is invariant under the action of X and Z. We codify this idea into the following key lemma, which is used in the proof of Theorem C. We state it in a simplified form, which is not correct as stated but captures the spirit of a special case. See Lemma 2.11 for the full and correct statement.

▶ **Lemma D** (Oversimplified version of Lemma 2.11). *Suppose X, Y, Z are three Lie subgroups of $\mathrm{GL}_n(\mathbb{C})$ that satisfy the TPP. If Y'_ε is a finite set of ε-parametrized families in Y, and there is a polynomial $p_\varepsilon(g)$ that is invariant under left multiplication by X and right multiplication by Z, such that*

$$p_\varepsilon(g) = \begin{cases} 1 + O(\varepsilon) & \text{if } g = I \\ 0 + O(\varepsilon) & \text{if } g \in (Y'_\varepsilon)^{-1}Y'_\varepsilon \setminus \{I\} \end{cases}$$

then there are finite subsets $X'_\varepsilon, Y'_\varepsilon$ of ε-parametrized elements of X and Z, of sizes $q^{\dim X}$ (resp., $q^{\dim Z}$) and border-separating polynomials for $(X'_\varepsilon, Y'_\varepsilon, Z'_\varepsilon)$ of degree $\deg p + O(q)$.

Thus, once we have a TPP construction of the "right" dimension (e. g., according to Theorem B), Lemma D implies that instead of finding appropriate finite subsets of X, Y, Z and a *set* of separating polynomials, to get $\omega = 2$ all we have to do is find an appropriate finite subset of Y and a *single* polynomial p as in the lemma, whose degree is $O(q)$.

When X and Z are well-studied subgroups, the ring of their invariant polynomials is often well understood, and this represents a significant simplification of the construction problem. The full detail of Lemma 2.11 is more complicated, but the additional complications in the statement of the lemma give us even further simplifications of the construction problem, which we take advantage of in our constructions in the running example and in Section 3. We also expect the techniques of this lemma to have further uses for additional constructions in the future.

1.5 Paper outline

In Section 2.1 we give the necessary definitions and proof of Theorem A, and in Section 2.2 we give its border rank version. In Section 2.3 we discuss the needed background on the representation theory of $\mathrm{GL}_n(\mathbb{C})$ – much of which we can use in a black-box fashion – and prove Theorem B. In Section 2.4 we show how to combine invariant theory, border rank, and Lie algebras to simplify the construction task; the key here is Lemma 2.11. In Section 3 we use the preceding ideas to give the construction that proves Theorem C. Finally, in Section 4 we conclude with our outlook and several open problems suggested by our framework.

Throughout the development of the framework in Section 2, we continue the example from Theorem 1.2 as a running example, and we show how each piece of the framework can be realized in that case.

The full version of the paper contains complete proofs of all claims [10].

2 A group-theoretic framework for infinite groups

In this section we describe the group-theoretic framework for obtaining matrix multiplication algorithms in infinite groups. The basic framework is in the next subsection, followed by a border-rank version for the important case of Lie groups. Subsection 2.3 proves some key bounds on the dimensions of irreducible representations of GL_n and U_n, which are the containing groups for our constructions in this paper. Finally subsection 2.4 describes machinery that reduces the main design task to finding a single separating polynomial in an invariant ring. These components come together in Section 3, where we give a construction achieving optimal degree separating polynomials.

2.1 Algorithms from the TPP in infinite groups

Let G be a group – not necessarily finite – with finite subsets X, Y, Z satisfying the TPP. Then we can embed matrix multiplication into the group algebra $\mathbb{C}[G]$ as in the case of finite groups. Note that $\mathbb{C}[G]$, where G may be infinite, consists of formal sums $\sum_{g\in G} \alpha_g g$, but now where each such sum has at most finitely many nonzero terms. Multiplication is as in the finite case.

To multiply a complex matrix A of size $|X| \times |Y|$ by a complex matrix B of size $|Y| \times |Z|$, we follow the approach described in the introduction. Indexing A by $X \times Y$ and B by $Y \times Z$, we define elements

$$\overline{A} = \sum_{x\in X, y\in Y} A[x,y](xy^{-1}) \quad \text{and} \quad \overline{B} = \sum_{y\in Y, z\in Z} B[y,z](yz^{-1})$$

of $\mathbb{C}[G]$ and observe that, as in the finite group case, the TPP implies that

$$\overline{A} \cdot \overline{B} = \sum_{x \in X, z \in Z} (AB)[x,z](xz^{-1}) + E, \tag{2.1}$$

where $E \in \mathbb{C}[G]$ is supported on $XY^{-1}YZ^{-1} \setminus XZ^{-1}$.

However, as the group algebra is no longer finite-dimensional when G is infinite, it is not immediately clear that the multiplication in the group algebra expressed in (2.1) can be carried out by a finite algorithm, even despite the fact that all the sums involved are themselves finite. One of our key new innovations is to introduce a new viewpoint on such a construction that will enable us to get a finite bilinear algorithm out of the above construction.

▶ **Definition 2.1.** *Given subsets $X, Y, Z \subseteq G$ of a group G, a* set of separating functions for (X, Y, Z) *is a collection of functions $\{f_{x,z} \colon G \to \mathbb{C} \mid x \in X, z \in Z\}$ such that*

$$f_{x,z}(g) = \begin{cases} 1 & \text{if } g = xz^{-1}, \text{ and} \\ 0 & \text{if } g \in XY^{-1}YZ^{-1} \setminus \{xz^{-1}\}. \end{cases}$$

Such functions always exist, but for them to be useful in matrix multiplication algorithms we need them to be in some sense "simple." To formulate the relevant notion – in which "simple" will ultimately imply "computable by smaller matrix multiplications" – we recall a standard definition from representation theory. Given a finite-dimensional representation $\rho \colon G \to \mathrm{GL}_n(\mathbb{C})$, a *representative function* of G associated to ρ (see, e. g., Procesi's book [23, §8.2][1]) is any function $G \to \mathbb{C}$ that is in the linear span of the functions $\{(g \mapsto \rho(g)_{i,j}) \mid i, j \in [n]\}$. Note that if ρ, ρ' are equivalent representations, they have the same set of representative functions. If $\mathcal{I}$ is a set of representations, we write $\mathrm{RepFun}(\mathcal{I})$ for the $\mathbb{C}$-linear span of all the representative functions of the representations in $\mathcal{I}$.

▶ **Theorem 2.2.** *Let G be a group (not necessarily finite), with finite subsets X, Y, Z satisfying the TPP. If R_{sep} is a finite set of finite-dimensional complex representations of G such that $\mathrm{RepFun}(R_{\mathrm{sep}})$ contains a set of separating functions for (X, Y, Z), then*

$$(|X||Y||Z|)^{\omega/3} \leq \sum_{\rho \in R_{\mathrm{sep}}} (\dim \rho)^{\omega}.$$

In particular, for $D = \sum_{\rho \in R_{\mathrm{sep}}} (\dim \rho)^2$ and $d_{max} = \max\{\dim \rho : \rho \in R_{\mathrm{sep}}\}$,

$$(|X||Y||Z|)^{\omega/3} \leq D \cdot d_{max}^{\omega-2}.$$

If G is a finite group, we may take $R_{\mathrm{sep}} = \mathrm{Irr}(G)$, the set of all irreducible representations of G, as $\mathrm{RepFun}(\mathrm{Irr}(G))$ is the collection of *all* functions $G \to \mathbb{C}$ when G is finite. In that case, we also have $D = |G|$, recovering the original theorem for finite groups [14, Theorem 4.1] as a special case of Theorem 2.2.

[1] The notion of representative function we use here is fairly standard; if one consults Procesi's book, one will see that his book works in the setting where G is a topological group, and requires the representations involved to be continuous, but the definition we use here works just as well. When G is a Lie group, as in our constructions later in the paper, the representations we use will in fact be continuous in the natural manifold topology on G rather than the discrete topology.

▶ Remark 2.3. In families of groups $\{G_i\}$ with "rapidly growing" irreducible representation dimensions, this theorem exhibits an "all-or-nothing" phenomenon, which we explain here. For the constructions in this paper and other natural constructions in classical Lie groups, $d_{max} \geq D^{1/2-g(i)}$ for some $g \in o(1)$ (and this is what we mean by "rapidly growing irreducible representation dimensions"). If we let $V = (|X||Y||Z|)^{1/3}$, it is clear that we must have $V > d_{max}$ for the inequality to imply *any* upper bound on ω. Together with the observation that d_{max} approaches $D^{1/2}$, this means that to prove any upper bound on ω we must have $V \geq D^{1/2-f(i)}$ for some $f \in o(1)$. The inequality from the theorem then becomes

$$D^{(1/2-f(i))\omega} \leq V^\omega \leq D \cdot d_{max}^{\omega-2} \leq D \cdot D^{(1/2-g(i))(\omega-2)}.$$

Taking the base-D logarithm of both sides, we get

$$\omega/2 - f(i)\omega \leq \omega/2 - g(i)\omega + 2g(i),$$

which implies $\omega \leq 2\left(\frac{g(i)}{g(i)-f(i)}\right)$. In constructions the natural thing to do is to ensure $f(i)$ goes to zero more rapidly than $g(i)$, in which case $\omega = 2$. While it is *possible* that $f(i)$ could approach $cg(i)$ for some $c > 0$ and yield an upper bound on ω strictly between 2 and 3, this would require detailed knowledge of the lower order terms of d_{max} and/or very fine control of the lower order terms of V, which we do not typically have.

Proof. Let X, Y, Z, R_{sep} be as in the statement, and let $\{f_{x,z} \mid x \in X, z \in Z\}$ be the claimed set of separating functions contained in $\text{RepFun}(R_{\text{sep}})$. For $f\colon G \to \mathbb{C}$, let $\overline{f}\colon \mathbb{C}[G] \to \mathbb{C}$ denote its unique linear extension to the group ring $\overline{f}(\sum \alpha_g g) := \sum \alpha_g f(g)$; as these sums have only finitely many nonzero terms by definition of the group ring (even when G is infinite), there is no issue of convergence. Applying $\overline{f_{x,z}}$ to (2.1) gives

$$\overline{f_{x,z}}(\overline{A} \cdot \overline{B}) = \sum_{x' \in X, z' \in Z} (AB)[x', z'] f_{x,z}(x'(z')^{-1}) + \overline{f_{x,z}}(E) = (AB)[x, z] + 0, \tag{2.2}$$

for $f_{x,z}$ is zero on all group elements of $X(Y)^{-1}Y(Z)^{-1}$ other than xz^{-1}, and (2.1) is entirely supported on $X(Y)^{-1}Y(Z)^{-1}$. To turn this into a finite algorithm, we will show that we can essentially do exactly the application of $\overline{f_{x,z}}$ in (2.2), but working only in the representations in R_{sep} rather than working in the full group ring.

For a representation $\rho \in R_{\text{sep}}$, let $\overline{\rho}$ denote the unique linear extension of ρ to $\mathbb{C}[G]$: $\overline{\rho}(\sum \alpha_g g) = \sum \alpha_g \rho(g)$, and let $\rho_{i,j}(g)$ be the (i,j) entry of the matrix $\rho(g)$, which we think of as a function $\rho_{i,j}\colon G \to \mathbb{C}$. Since $f_{x,z}$ is in $\text{RepFun}(R_{\text{sep}})$ by assumption, we can write $f_{x,z}$ as a $\mathbb{C}$-linear combination of the functions $\{\rho_{i,j} \mid \rho \in R_{\text{sep}},\ i,j \in [\dim \rho]\}$, say $f_{x,z} = \sum_{\rho \in R_{\text{sep}}} \sum_{i,j \in [\dim \rho]} M_{x,z,i,j} \rho_{i,j}$. Then we define $\widehat{f_{x,z}}(\rho)$ to be the matrix $M_{x,z,*,*}$, i.e.,

$$f_{x,z}(g) = \sum_{\rho \in R_{\text{sep}}} \sum_{i,j \in [\dim \rho]} \widehat{f_{x,z}}(\rho)_{i,j} \rho_{i,j}(g)$$

for all $x \in X, z \in Z, g \in G$. Finally, extending linearly and applying $\overline{f_{x,z}}$ to $\overline{A} \cdot \overline{B}$ as in (2.2), we get

$$\begin{aligned}
(AB)[x,z] &= \overline{f_{x,z}}(\overline{A} \cdot \overline{B}) \\
&= \sum_{\rho \in R_{\text{sep}}} \sum_{i,j \in [\dim \rho]} \widehat{f_{x,z}}(\rho)_{i,j} \overline{\rho_{i,j}}(\overline{A} \cdot \overline{B}) \\
&= \sum_{\rho \in R_{\text{sep}}} \langle \widehat{f_{x,z}}(\rho), \overline{\rho}(\overline{A} \cdot \overline{B}) \rangle \\
&= \sum_{\rho \in R_{\text{sep}}} \langle \widehat{f_{x,z}}(\rho), \overline{\rho}(\overline{A}) \cdot \overline{\rho}(\overline{B}) \rangle
\end{aligned}$$

The summation and inner product are linear functions whose coefficients are independent of the input matrices A, B, so they are "free" in a bilinear algorithm.

The product $\overline{\rho}(\overline{A}) \cdot \overline{\rho}(\overline{B})$ is a product of $d_\rho \times d_\rho$ matrices, where $d_\rho = \dim \rho$. Hence, the bilinear (i.e., tensor) rank of the preceding expression gives the following bound. Using rk to denote the tensor rank, and $\langle n, m, p \rangle$ to denote the tensor corresponding to matrix multiplication of an $n \times m$ matrix times and $m \times p$ matrix, we get

$$\mathrm{rk}\,\langle |X|, |Y|, |Z| \rangle \leq \sum_{\rho \in R_{\mathrm{sep}}} \mathrm{rk}\,\langle d_\rho, d_\rho, d_\rho \rangle.$$

Exactly as in the finite group case [14], by symmetrizing we effectively get a square matrix multiplication of size $(|X|\,|Y|\,|Z|)^{1/3}$ on the left side, and by the tensor power trick the right side here can be replaced by $\sum_{\rho \in R_{\mathrm{sep}}} (\dim \rho)^\omega$. For the last sentence of the theorem statement, we have $\sum_{\rho \in R_{\mathrm{sep}}} d_\rho^\omega \leq \sum d_\rho^2 d_{max}^{\omega-2} = d_{max}^{\omega-2} \cdot D$. ◀

▶ Remark 2.4. The image $\overline{\rho}(\mathbb{C}[G])$ is the full $d_\rho \times d_\rho$ matrix ring if and only if ρ is an irreducible representation. In particular, although we will not take advantage of this in the present paper, we note that when ρ is not irreducible, we can replace $\mathrm{rk}\,\langle d_\rho, d_\rho, d_\rho \rangle$ (or d_ρ^ω) with the tensor rank of multiplying matrices in the image of $\overline{\rho}$, which may only be a subspace of all matrices. (This was true in the case of finite groups as well, but over $\mathbb{C}$ for finite groups we can use irreducible representations without loss of generality. For finite groups and representations in characteristic dividing $|G|$ this no longer holds, and for infinite groups even in characteristic zero it need not hold.)

2.2 Border rank version in Lie groups

For this section, let G be a (real or complex) Lie group. This includes familiar groups such as $\mathrm{GL}_n(\mathbb{R})$, $\mathrm{GL}_n(\mathbb{C})$, the orthogonal group O_n, and the unitary group U_n. Indeed, these examples will be the primary groups we use in our constructions later in the paper, although the framework laid out in this section is by no means limited to these particular examples.

For the purposes of this section, by a *1-parameter family* of elements of G we mean an analytic function $x\colon (0, \alpha) \to G$ for some $\alpha > 0$. If $X = \{x_1, \ldots, x_k\}$ is a collection of 1-parameter families x_i and ε is in their domain of definition, then we write $X(\varepsilon) = \{x_1(\varepsilon), \ldots, x_k(\varepsilon)\}$, which is just a finite subset of G.

▶ **Definition 2.5** (Border-separating functions). *Given sets X, Y, Z of 1-parameter families of elements of G with domain of definition $(0, \alpha)$, a* set of border-separating functions for (X, Y, Z) *is a collection of analytic functions $\{f_{x,z}\colon G \times (0, \alpha) \to \mathbb{C} \mid x \in X, z \in Z\}$ such that, for $0 < \varepsilon < \alpha$,*

$$f_{x,z}(g, \varepsilon) = \begin{cases} 1 + O(\varepsilon) & \text{if } g = x(\varepsilon) z(\varepsilon)^{-1}, \text{ and} \\ 0 + O(\varepsilon) & \text{if } g \in X(\varepsilon) Y(\varepsilon)^{-1} Y(\varepsilon) Z(\varepsilon)^{-1} \setminus \{x(\varepsilon) z(\varepsilon)^{-1}\}. \end{cases}$$

Here, as is standard for analytic functions of a small variable ε, the big-Oh notation means asymptotically as $\varepsilon \to 0$.

Now, we take our representative functions to come from *analytic* representations of our Lie group, and, as in most things related to border rank, allow ourselves 1-parameter families of representative functions. Let $\mathbb{C}^{(0,\alpha)}$ denote the set of analytic functions $(0, \alpha) \to \mathbb{C}$. Given a set $\mathcal{I}$ of analytic representations of G, we write $\mathrm{RepFun}_{\mathrm{fam}}(\mathcal{I})$ for the $\mathbb{C}^{(0,\alpha)}$-linear span of all the representative functions associated to any $\rho \in \mathcal{I}$.

Given three sets X, Y, Z of 1-parameter families of elements of G, we say they satisfy the TPP if $X(\varepsilon), Y(\varepsilon), Z(\varepsilon)$ satisfy the TPP for all ε in their domain of definition. Given such X, Y, Z, we may encode ε-approximate matrix multiplication (à la border rank) into the group ring $\mathbb{C}[G]$ in a similar manner as before, but now parameterized by $\varepsilon \in (0, \alpha)$. For an $|X| \times |Y|$ matrix A and $|Y| \times |Z|$ matrix B, we define the following functions $(0, \alpha) \to \mathbb{C}[G]$:

$$\overline{A}(\varepsilon) = \sum_{i\in[n], j\in[m]} A[i,j](x_i(\varepsilon)y_j(\varepsilon)^{-1}) \quad \text{and} \quad \overline{B}(\varepsilon) = \sum_{j\in[m], k\in[\ell]} B[j,k](y_j(\varepsilon)z_k(\varepsilon)^{-1}).$$

Then

$$\overline{A}(\varepsilon) \cdot \overline{B}(\varepsilon) = \sum_{i\in[n], k\in[\ell]} (AB)[i,k](x(\varepsilon)z(\varepsilon)^{-1}) + E(\varepsilon), \tag{2.3}$$

where $E(\varepsilon) \in \mathbb{C}[G]$ is supported on $X(\varepsilon)Y(\varepsilon)^{-1}Y(\varepsilon)Z(\varepsilon)^{-1} \setminus X(\varepsilon)Z(\varepsilon)^{-1}$.

▶ **Theorem 2.6.** *Let G be a* ***Lie***[2] *group, with finite sets X, Y, Z of* ***1-parameter families*** *satisfying the TPP. If R_{sep} is a finite set of finite-dimensional* ***analytic*** *representations of G such that $\text{RepFun}_{\boldsymbol{fam}}(R_{\text{sep}})$ contains a set of* ***border-****separating functions for (X, Y, Z), then*

$$(|X|\,|Y|\,|Z|)^{\omega/3} \leq \sum_{\rho\in R_{\text{sep}}} (\dim \rho)^{\omega}.$$

In particular, for $D = \sum_{\rho\in R_{\text{sep}}} (\dim \rho)^2$, and $d_{max} = \max\{\dim \rho : \rho \in R_{\text{sep}}\}$,

$$(|X|\,|Y|\,|Z|)^{\omega/3} \leq D \cdot d_{max}^{\omega-2}.$$

The proof is very similar to Theorem 2.2, but using border rank instead of rank, so we postpone it to the full version of the paper [10].

An appealing way to use the freedom of border rank is to first find a TPP construction of Lie subgroups X, Y, Z, and then to define finite subsets of those three using their Lie algebras, viz.:

$$X' = \{\exp(\varepsilon a) : a \in A, \text{ some finite subset of the Lie algebra of } X\},$$

where $\varepsilon \to 0$ is the parameter we use for border rank. We will in fact show how to do this in a fairly generic way in Lemma 2.11 below, using some additional machinery that we develop first.

2.3 From representations to degree bounds in GL_n and U_n

In this paper, our constructions will all take place in GL_n or (slight variants of) the unitary group U_n, although the framework of Theorems 2.2 and 2.6 is much more general. These groups have some useful properties that will motivate some of the machinery we develop below – even though that machinery will also end up being more general – so we take a brief interlude to highlight the relevant properties of GL_n, before returning to the general abstract framework in the next section.

For both GL_n and U_n, there is a natural correspondence between representations and polynomials of a given degree. By focusing on separating polynomials (instead of more general separating functions), this correspondence will allow our constructions to focus only

[2] We have put in **bold** the parts of the statement of Theorem 2.6 that differ from Theorem 2.2.

on the *degree* of the separating polynomials. The upshot is that instead of thinking directly about what representations will comprise R_{sep}, we can focus solely on the degree of our separating polynomials when working in these groups.

In the rest of this section we review the relevant aspects of the representation theory of GL_n and U_n, and extract from those the relevant target bounds on degree to get bounds on ω. A standard reference for the representation theory of these groups is [19]. It is a standard fact in representation theory that the finite-dimensional representation theory of these two groups are essentially the same, and everything we say in this section will apply to both groups.

The irreducible polynomial representations ρ of $\mathrm{GL}_n(\mathbb{C})$ and U_n – where the entries of the matrix $\rho(g)$ are polynomials in the entries of the matrix g – are indexed by integer partitions $\lambda = (\lambda_1, \ldots, \lambda_n)$ with at most n parts, where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$. The matrix coefficients of the irreducible representations indexed by partitions of s span the functions from G to $\mathbb{C}$ that are expressible as degree s polynomials in the entries of GL_n. We write $\mathrm{Irr}_s(G)$ to denote the set of (pairwise non-isomorphic) irreducible representations of G indexed by partitions of the integer s.

The following bound will determine the target degree of the separating polynomials in our constructions, as in Corollary 2.8 below.

▶ **Lemma 2.7.** *Let $n \geq 3$ and $s \geq 2$. For $\rho \in \mathrm{Irr}_s(\mathrm{GL}_n)$ (resp., $\mathrm{Irr}_s(\mathrm{U}_n)$), $\dim(\rho) \leq s^{\binom{n}{2}}$.*

Although the result follows from some simple asymptotic analysis of standard results about the representation theory of GL_n, we could not find it in the literature, so we provide a full proof in [10].

Here and below it is helpful to think of n as large but fixed, and s growing independently of n.

▶ **Corollary 2.8.** *Suppose $X, Y, Z \subseteq \mathrm{GL}_n(\mathbb{C})$ (or U_n) satisfy the TPP, and there is a set of separating polynomials for (X, Y, Z) of degree at most s. Then ω satisfies*

$$(|X|\,|Y|\,|Z|)^{\omega/3} \leq s^{\binom{n}{2}(\omega-2)} \cdot \binom{s+n^2}{n^2}.$$

In particular, if $|X|, |Y|, |Z| \geq s^{(1-o_s(1))(n^2/2-o_n(n))}$, then $\omega = 2$.

The same holds if X, Y, Z are 1-parameter families that satisfy the TPP and we replace separating polynomials with border-separating polynomials.

Equivalently, if $|X|, |Y|, |Z| \geq q^{n^2/2-o_n(n)}$ and there are (border-)separating polynomials of degree at most $q^{1+o_q(1)}$, then $\omega = 2$. See the full version of the paper for a proof [10].

Next we show that the Lie TPP construction of [14] (Theorem 1.2 above) in fact admits separating polynomials of degree $O(q^2)$. In Section 2.4, we will show how to use a border-rank version of our framework to construct *finite* sets X, Y, Z, whose size we can then compare to the degree of the separating polynomials as in Corollary 2.8.

Separating polynomials for the running example in $\mathbf{GL}_n(\mathbb{R})$

We begin by describing a finite subset Y_q of the orthogonal group O_n, with the property that the diagonal entries of matrices in the quotient set $Q(Y_q)$ have a limited number of possible values, in the sense formalized in the following lemma. Here we treat n as fixed, so the implicit constant in $O(q^2)$ can depend on n.

▶ **Lemma 2.9.** *For each positive integer q, there exists a subset $Y_q \subseteq \mathrm{O}_n$ of cardinality at least $q^{n^2/2-(5/2)n}$, with the following property: for all $y, y' \in Y_q$, if y and y' agree in their first i columns, then*

$$(y^T y')[i+1, i+1] \in W_q,$$

where $W_q \subseteq \mathbb{R}$ is a fixed finite set of cardinality $O(q^2)$.

For a proof, see the full version of the paper [10].

We remark that one might hope to improve the cardinality of W_q to $O(q^{2-\varepsilon})$ by replacing V_i with a more cleverly chosen set of unit vectors with fewer distinct pairwise inner products. This would lead to an improved bound on the degree of the separating polynomials described in the following theorem, and if one could take $\varepsilon = 1$ this would yield separating polynomials of optimal degree. Unfortunately, this is impossible by the solution to the high-dimensional version of the Erdős distinct distances problem [25].

▶ **Theorem 2.10.** *Let G, X, Y, Z be as in Theorem 1.2, let $X_q \subseteq X$ be those lower unitriangular matrices whose below-diagonal entries are integers in $\{1, 2, \ldots, q\}$, let $Z_q \subseteq Z$ be those upper unitriangular matrices whose above-diagonal entries are integers in $\{1, 2, \ldots, q\}$, and let $Y_q \subseteq Y$ be the set of orthogonal matrices guaranteed by Lemma 2.9. Then there are separating polynomials for X_q, Y_q, Z_q of degree $O(q^2)$.*

For a proof, see the full version of the paper [10].

Note that here we have $\dim X = \dim Y = \dim Z = (n^2 - n)/2$, but for Corollary 2.8 we would need them to have dimension $n^2/2 - o_n(n)$, and the separating polynomial we get has degree $O_q(q^2)$, but Corollary 2.8 needs degree at most $q^{1+o_q(1)}$. In Section 3 we give a construction of the same relative dimension (namely, $\frac{\dim G}{2} - \Theta(n)$) but with separating polynomials of degree $O(q)$, meeting the degree bound of Corollary 2.8.

2.4 Separating polynomials from a single invariant polynomial

In this section, we return to the case of a general matrix Lie group $G \subseteq \mathrm{GL}_n(\mathbb{C})$ or $\mathrm{GL}_n(\mathbb{R})$, and we show how to leverage invariant theory to construct separating polynomials whose degree is not too large, and that thus have a hope of meeting the degree bound of Corollary 2.8 in the case of $G = \mathrm{GL}_n(\mathbb{C})$ or $G = \mathrm{U}_n$.

The key result in this section, Lemma 2.11, combines border rank and Lie algebras, as suggested at the end of Section 2.2, with a new ingredient from invariant theory. The lemma lets us "split" the construction of sets satisfying the TPP and (border-)separating polynomials into two parts: the first part only involves the first and third sets X and Z (essentially, a separating polynomial version of the "double product property" (DPP), which is that $x^{-1}x'z^{-1}z' = 1$ iff $x = x'$ and $z = z'$)), and the second part involves only the middle set Y and the ring of XZ-invariant polynomials. From one view, this lets us start with a Y, and "work backwards," shifting the design task onto X, Z, and their invariant polynomials.

More specifically, Lemma 2.11 instantiates the following idea. Roughly, we would like to construct the separating polynomial $f_{x,z}$ as a product of three factors: one that is the indicator function of x among the set X (1 on x, zero on all other elements of X), one that is the indicator function of z^{-1} among Z^{-1}, and one on that is the indicator function p_0 of I among $Y^{-1}Y$. However, the issue here is that $f_{x,z}$, and hence these factors, does not really have separate access to these three parts: it only gets as input the product $xy^{-1}y'z^{-1}$. Our initial motivation to use invariant polynomials was that if p_0 has the property that $p_0(xy^{-1}y'z^{-1}) = p_0(y^{-1}y')$ for all $x \in X, z \in Z$, then it in fact *does* get direct access to only the Y-part of the input. Lemma 2.11 then gives generic machinery that, from such an XZ-invariant p_0, constructs a full set of (border-)separating polynomials, which have the correct degree if p_0 has the correct degree.

One advantage of this approach, in addition to separating out the design task associated with Y more from the specification of X and Z, is that it only necessitates the construction of at most three polynomials p_0, p_X, p_Z, rather than a whole set $\{f_{x,z}\}$ of $|X|\,|Z|$-many polynomials, and the p_X, p_Z polynomials are usually easy to construct if X and Z satisfy the DPP.

We now proceed with the technical details. By a *polynomial* on a matrix Lie group $G \subseteq \mathrm{GL}_n(\mathbb{C})$ or $G \subseteq \mathrm{GL}_n(\mathbb{R})$, we mean a function $G \to \mathbb{C}$ that can be expressed as a polynomial in the n^2 matrix entries x_{ij} of elements of G, with complex coefficients,[3] and similarly for a polynomial on the Lie algebra $\mathrm{Lie}(G) \subseteq M_n(\mathbb{C})$. By a 1-parameter family of polynomials we mean a polynomial in the n^2 matrix entries x_{ij} of elements of G, with coefficients in $\mathbb{C}^{(0,\alpha)}$ for some $\alpha > 0$.

Because we will use it frequently, we introduce the notation

$$\mathrm{Inv}_{X,Z} = \{p \in \mathbb{C}[M_{ij} : i,j \in [n]] \mid p(xMz) = p(M) \text{ for all } x \in X, z \in Z\}.$$

It is readily observed that $\mathrm{Inv}_{X,Z}$ is closed under multiplication and addition, and hence is a subring of the polynomial ring. In favorable (and many well-studied) cases, this subring is finitely generated over $\mathbb{C}$ (see, e. g., [20, 17]), and when a finite generating set of $\mathrm{Inv}_{X,Z}$ is known, we can focus on designing a polynomial that is composed with those generating invariants – which is then automatically invariant itself – in order to obtain p_0. We will see an example of this below.

In a Lie group G, for convenience let us say a subset $X \subseteq G$ is a *Lie submanifold* if X is a submanifold containing the identity of G, and such that for every v in the tangent space of X at the identity, for all sufficiently small $\varepsilon > 0$, the exponential $\exp(\varepsilon t)$ is in X. If X is a Lie submanifold, we use $\mathrm{Lie}(X)$ to denote the tangent space to X at the identity, even though this need not be a Lie algebra.

▶ **Lemma 2.11** (Splitting separating functions into invariant functions and double-product property)**.** *Let G be a matrix Lie subgroup of $\mathrm{GL}_n(\mathbb{F})$ for $\mathbb{F} \in \{\mathbb{R}, \mathbb{C}\}$, with Lie submanifolds X, Z and a finite set Y of 1-parameter families. Let q be a positive integer.*

[3] Warning: for readers familiar with algebraic geometry, if G is an algebraic group our definition here is *not* quite the same thing as an element of the coordinate ring of G. For example, when $G = \mathrm{U}_n$, we view G as a subset of $\mathrm{GL}_n(\mathbb{C})$ and want to consider polynomials only in the n^2 matrix entries of these complex matrices, but as an algebraic group G is a *real* algebraic group, with twice as many (real) coordinates, and this difference of a factor of 2 can actually be important. We believe there is a modification of our framework that works in the setting of algebraic groups, but leave that for future work.

If there exist

- *continuous functions* $f_X\colon \mathbb{F}^{d_X} \to \mathrm{Lie}(X)$ *and* $f_Z\colon \mathbb{F}^{d_Z} \to \mathrm{Lie}(Z)$*, and polynomials* $p_X\colon \mathrm{Lie}(G) \to \mathbb{F}^{d_X}$ *and* $p_Z\colon \mathrm{Lie}(G) \to \mathbb{F}^{d_Z}$ *such that*

$$p_X(f_X(v) - f_Z(v')) = v \qquad \text{and} \qquad p_Z(f_X(v) - f_Z(v')) = v',$$

for all $v \in \mathbb{F}^{d_X}, v' \in \mathbb{F}^{d_Z}$,[4]

and

- *a 1-parameter family of polynomials* $p_0(g,\varepsilon) \in \mathrm{Inv}_{X,Z}$ *such that*

$$p_0(g,\varepsilon) = \begin{cases} 1 + O(\varepsilon) & \text{if } g = I \\ 0 + O(\varepsilon) & \text{if } g \in Y(\varepsilon)^{-1}Y(\varepsilon) \setminus \{I\}, \end{cases}$$

then there exist

- *a reparametrization* Y' *of* Y *and finite sets* X', Z' *of 1-parameter families in* X*,* Z *(resp.) such that* $X'(\varepsilon), Y'(\varepsilon), Z'(\varepsilon)$ *satisfy the TPP for all* ε *in some non-empty range* $(0, \alpha')$*,* $|X'| \geq q^{d_X}$*, and* $|Z'| \geq q^{d_Z}$*, and*
- *a set of border-separating polynomials for* (X', Y', Z') *of degree*[5] *at most* $\deg p_0 + O((d_X + d_Z)q)$*.*

As the first condition of Lemma 2.11 may be a little intimidating, before the proof we give an example to help allay such fears:

▶ **Lemma 2.12.** *If* $X, Z \leq G \leq \mathrm{GL}_n(\mathbb{R})$ *are Lie subgroups that intersect trivially, then there exist* f_X, f_Z, p_X, p_Z *satisfying the first condition of Lemma 2.11, with* $d_X = \dim X$ *and* $d_Z = \dim Z$*.*

Note that if we drop the requirement that p_0 be in $\mathrm{Inv}_{X,Z}$, then the hypotheses of Lemma 2.11 would be almost trivially satisfied whenever X and Z have trivial intersection. In applying Lemma 2.11, it is only the use of invariants that adds any real difficulty; however, the use of invariants is also a lynchpin to reach the conclusion.

For the proofs of Lemmas 2.11 and 2.12. see the full version of the paper [10].

Continuing our running example, we now show how to satisfy the hypothesis of Lemma 2.11, and thus get a TPP construction using border rank, invariant polynomials, and Lie algebras.

Example of a separating polynomial in the invariant ring

We return to the example in $\mathrm{GL}_n(\mathbb{R})$ described in Theorem 1.2. The following lemma is straightforward to check. Let lpm_i denote the polynomial that is the i-th leading principal minor of an $n \times n$ matrix; that is, lpm_i is the determinant of the upper-left $i \times i$ submatrix of an $n \times n$ matrix.

▶ **Lemma 2.13.** *Let* $X, Z \subseteq \mathrm{GL}_n(\mathbb{R})$ *be the sets of lower unitriangular matrices and the upper unitriangular matrices, respectively. Then* $\mathrm{Inv}_{X,Z}$ *contains the leading principal minors.*

---

[4] Notice that, since p_X, p_Z essentially invert f_X, f_Z on the sum of their images, this condition implies that $\mathrm{Span}_{\mathbb{F}}(\mathrm{Image}(f_X)) \cap \mathrm{Span}_{\mathbb{F}}(\mathrm{Image}(f_Z)) = 0$, a DPP-like condition.

[5] Degree as measured only as a function of the matrix entries – ε is still considered a separate parameter. Equivalently, degree as polynomials with coefficients in $\mathbb{C}^{(0,\alpha)}$.

In fact, in this case $\mathrm{Inv}_{X,Z}$ is generated by the leading principal minors, but we will not need this stronger fact for our constructions.

▶ **Theorem 2.14.** *Let G, X, Y, Z be as in Theorem 1.2. For any $q > 0$, there is a set $\widetilde{Y}$ of 1-parameter families taking values in Y, of size $q^{\dim Y}$, as well as functions f_X, f_Z, polynomials p_X, p_Z, and a 1-parameter family of polynomials p_0 of degree $O(q^2)$ satisfying the hypotheses of Lemma 2.11 with $d_X = \dim X$, $d_Z = \dim Z$, and $\widetilde{Y}$ replacing Y.*

Consequently, there is a TPP construction with sizes $q^{\dim X}, q^{\dim Y}, q^{\dim Z}$, resp., admitting border-separating polynomials of degree $O(q^2)$.

For a proof, see the full version of the paper [10].

3 Separating polynomials of degree $O(q)$

In this section we give a construction, now in a non-compact special unitary group, in which the Lie subgroups X, Y, Z satisfy the TPP and have dimension approaching half the ambient dimension. We show the existence of finite sets $X_q \subseteq X$, $Y_q \subseteq Y$, and $Z_q \subseteq Z$ satisfying $|X_q| = q^{\dim X}$, $|Y_q| = q^{\dim Y}$, $|Z_q| = q^{\dim Z}$ and with separating polynomials of degree $O(q)$. As shown in Corollary 2.8, this degree bound is the quantitative requirement on the degree for achieving exponent 2; where the construction falls short is that it approaches half the dimension "just barely" too slowly, at a rate of $\Theta_n(n)$ rather than $o_n(n)$, and second, it is in a unitary group, so it approaches half the dimension of the unitary group, rather than half the dimension of $\mathrm{GL}_n(\mathbb{C})$.

3.1 The construction

We assume n is even and define the containing group

$$G = \mathrm{SU}_{n/2,n/2} = \{M \in \mathrm{GL}_n(\mathbb{C}) \mid M^* Q M = Q \text{ and } \det(M) = 1\},$$
$$\text{with } Q = \left(\begin{array}{c|c} I & \\ \hline & -I \end{array}\right).$$

Further, let J be the matrix with ones on the antidiagonal (in positions $(n+1-i, i)$),

$$U = \frac{1}{\sqrt{2}} \cdot \left(\begin{array}{c|c} I & J \\ \hline J & -I \end{array}\right),$$
$$D_0 = \mathrm{diag}(n, n-1, n-2, \ldots, n/2+1, (n/2+1)^{-1}, (n/2+2)^{-1}, \ldots, n^{-1}),$$

and

$$D = U D_0 U^*. \tag{3.1}$$

The only properties we will need from these matrices are that D_0 is diagonal, $D_0^* D_0$ has distinct, positive, real diagonal entries, and U is unitary such that $U D_0 U^*$ is in G, but we use these particular matrices for concreteness. In Lemma 3.6 we will show that D as in (3.1) is indeed in G.

The group $\mathrm{SU}_{n/2,n/2}$ is not compact, and it may be less familiar than the compact group SU_n, but it is equally useful for our purposes. Both $\mathrm{SU}_{n/2,n/2}$ and SU_n are subgroups of $\mathrm{SL}_n(\mathbb{C})$, and their irreducible representations are exactly the restrictions of those of $\mathrm{SL}_n(\mathbb{C})$. See the appendix of the full version [10] for a brief review of this topic.

The subgroup $\mathrm{U}_n \cap G$ is the block diagonal subgroup $U_{n/2} \times U_{n/2}$, and we need the following subspace, which is close to $\mathrm{Lie}(U_n \cap G)$ but is just a bit smaller:

$$S = \left\{ \left(\begin{array}{c|c} A_0 & 0 \\ \hline 0 & A_1 \end{array} \right) \mid A_0, A_1 \text{ are skew-Hermitian with zeros on the diagonal} \right\}.$$

▶ **Theorem 3.1.** *Let G, D, and S be as above. Let $\mathcal{I}_q \subset \mathbb{C}$ be the set of $a + ib$ with $a, b \in [-\lceil\sqrt{q}/2\rceil, \lceil\sqrt{q}/2\rceil]$. Define the following ε-parametrized subsets of G:*

$$X = \{D^{-1} \exp(\varepsilon A) D \mid A \in S\},$$
$$Z = \{D \exp(\varepsilon A) D^{-1} \mid A \in S\},$$
$$Y_q = \left\{ \exp(\varepsilon A) \mid A \in S, \text{ entries of } A \text{ lie in } \mathcal{I}_q \right\}.$$

Then there exist f_X, f_Z, p_X, p_Z, and p_0 of degree $O(q)$ satisfying the hypothesis of Lemma 2.11. Hence by Lemma 2.11, there exist three ε-parametrized subsets $X_q, Y_q, Z_q \subseteq G$, all of size at least $q^{n^2/4 - n/2}$, which satisfy the TPP and admit border-separating polynomials of degree $O(q)$.

3.2 A precursor construction in $\mathbf{GL}_n(\mathbb{C})$

In preparation for the proof of Theorem 3.1, we first establish a precursor in the containing group $\mathrm{GL}_n(\mathbb{C})$. Recall that the unitary group U_n is the set of matrices M for which $M^*M = I$, and it has half the dimension of the containing group.[6] We will shortly show that three conjugates of U_n in $\mathrm{GL}_n(\mathbb{C})$ almost satisfy the TPP (Theorem 3.3). The proof will make use of the following lemma:

▶ **Lemma 3.2.** *If $M \in \mathrm{U}_n$ and $D = UD_0U^*$ as defined above, then*

$$\mathrm{tr}(D^*M^*D^*DMD) \leq \mathrm{tr}((D^*D)^2)$$

with equality if and only if U^*MU is diagonal.

For a proof, see the full version of the paper [10].

This lemma gives a convenient way to prove that the three subgroups defined in the following theorem satisfy the TPP up to a "failure at the diagonal":

▶ **Theorem 3.3.** *Let $D = UD_0U^*$ be as defined as above. Then the three subgroups $X = D^{-1}\mathrm{U}_nD$, $Y = \mathrm{U}_n$, and $Z = D\mathrm{U}_nD^{-1}$ of $\mathrm{GL}_n(\mathbb{C})$ satisfy the following property. For all*

$$x = D^{-1}M_1D \in X, \quad y = M_2 \in Y, \quad z = DM_3D^{-1} \in Z,$$

if $xyz = I$, then U^*M_1U, U^*M_2U, and U^*M_3U are diagonal matrices.

For a proof, see the full version of the paper [10].

In the variant presented in the next section, we will form our TPP sets as exponentials of Lie algebra elements, and for this, we need a version of Lemma 3.2 that describes the deviation of $\mathrm{tr}(D^*M^*D^*DMD)$ below $\mathrm{tr}((D^*D)^2)$ explicitly as a function of the infinitesimal. Recall that the Lie algebra of U_n is the set of skew-Hermitian matrices.

[6] The unitary group is a real algebraic group, so we need to count real dimensions. The containing group has $2n^2$ real dimensions and the unitary group has n^2 real dimensions.

▶ **Lemma 3.4.** *Let $D = UD_0U^*$ be as defined above, with $d_1, d_2, \ldots, d_n$ being the diagonal entries of D_0. Let A, B be skew-Hermitian matrices and let $M = \exp(\varepsilon A)(\exp(\varepsilon B))^{-1}$. Then*

$$\operatorname{tr}(D^*M^*D^*DMD) = \operatorname{tr}((D^*D)^2) - \varepsilon^2 \sum_{i<j} (|d_i|^2 - |d_j|^2)^2 |C[i,j]|^2 + O(\varepsilon^3),$$

where $C = U^*(A - B)U$. In particular, this quantity is $\operatorname{tr}((D^*D)^2) + O(\varepsilon^3)$ if and only if C is diagonal.

For a proof, see the full version of the paper [10].

In preparation for the next subsection, we now describe some invariant polynomials in $\mathrm{Inv}_{X,Z}$ that we'll use below, where $X = D^{-1}\mathrm{U}_n D$ and $Z = D\mathrm{U}_n D^{-1}$. For an $n \times n$ matrix M and subsets $S, T \subseteq [n]$, we use $M_{S,T}$ to denote the $|S| \times |T|$ sub-matrix of M whose rows are indexed by the elements of S and whose columns are indexed by the elements of T.

▶ **Lemma 3.5.** *For each k, the function $p_k(M)$ defined as*

$$\sum_{\substack{S,T \subseteq [n] \\ |S|=|T|=k}} |\det((DMD)_{S,T})|^2$$

lies in $\mathrm{Inv}_{X,Z}$, where $X = D^{-1}\mathrm{U}_n D$ and $Z = D\mathrm{U}_n D^{-1}$.

We have been implicitly using this fact for p_1, which happens to be the complex Frobenius norm-squared function. In particular, $\operatorname{tr}(D^*M^*D^*DMD) = p_1(M)$. For a proof of Lemma 3.5, see the full version of the paper [10].

Note that because of the complex norm, these functions are not polynomials in the complex matrix entries (but they are polynomials in the entries of the natural real embedding into twice the dimension). In the next section we restrict the containing group to a special unitary group, which has the crucial side effect of making these functions polynomials in the *complex* matrix entries, because the complex conjugate of a given entry can be found as low-degree polynomial of the entries of the inverse matrix.

3.3 Proof of Theorem 3.1

Recall that the containing group of our construction in Theorem 3.1 is the following unitary group:

$$G = \mathrm{SU}_{n/2,n/2} = \{M \in \mathrm{GL}_n(\mathbb{C}) \mid M^*QM = Q \text{ and } \det(M) = 1\},$$
$$\text{with } Q = \left(\begin{array}{c|c} I & \\ \hline & -I \end{array}\right).$$

Our TPP triple in G stated in Theorem 3.1 is obtained by taking the "almost TPP" triple of Theorem 3.3 (recall that the construction almost satisfies the TPP except for a "failure at the diagonal") and intersecting it with G. In this section, we will fix the failure at the diagonal using Lemma 2.11, to get a genuine TPP construction and border-separating polynomials of the optimal degree.

Recall that our construction in G is given by

$$X = \{D^{-1}\exp(\varepsilon A)D \mid A \in S\},$$
$$Z = \{D\exp(\varepsilon A)D^{-1} \mid A \in S\},$$
$$Y_q = \{\exp(\varepsilon A) \mid A \in S, \text{ entries of } A \text{ lie in } \mathcal{I}_q\},$$

where $\mathcal{I}_q \subset \mathbb{C}$ is the set of $a + ib$ with $a, b \in [-\lceil\sqrt{q}/2\rceil, \lceil\sqrt{q}/2\rceil]$ and

$$S = \left\{ \left(\begin{array}{c|c} A_0 & 0 \\ \hline 0 & A_1 \end{array}\right) \mid A_0, A_1 \text{ are skew-Hermitian with zeros on the diagonal} \right\}.$$

We will prove Theorem 3.1 in three steps.

Step 1 summary: We show that the sets X and Z in the theorem statement are subsets of $D^{-1}\mathrm{U}_n D \cap G$ and $D\mathrm{U}_n D^{-1} \cap G$ and hence the ring of invariant polynomials contains the polynomials specified in Lemma 3.5. To do this we'll describe the Lie algebra of $U_n \cap G$, and show that D is in G.

Step 2 summary: Next, we describe the separating polynomial for Y_q in this invariant ring, and show that it has degree $O(q)$ as promised.

Step 3 summary: Finally we describe the functions f_X, f_Z, p_X, and p_Z and apply Lemma 2.11. For this we prove a version of the double product property for X and Z.

Step 1: Our TPP triple is defined relative to D, and we now show that the D defined in Section 3.1 is indeed in G as claimed. Recall that we use J for the matrix with ones on the antidiagonal $(n + 1 - i, i)$ and zeroes elsewhere (not the all-ones matrix).

▶ **Lemma 3.6.** *The matrix $D = UD_0U^*$ is an element of G, where*

$$U = \frac{1}{\sqrt{2}} \cdot \left(\begin{array}{c|c} I & J \\ \hline J & -I \end{array}\right),$$

and $D_0 = \mathrm{diag}(n, n-1, n-2, \ldots, n/2+1, (n/2+1)^{-1}, (n/2+2)^{-1}, \ldots, n^{-1})$.

See the full version for the proof [10].

Given that we will define our TPP triple by intersecting the three subgroups from the previous subsection with G, we start by describing the intersection of $Y = \mathrm{U}_n$ with G:

▶ **Lemma 3.7.**

$$\mathrm{U}_n \cap G = \left\{ \left(\begin{array}{c|c} M_{1,1} & 0 \\ \hline 0 & M_{2,2} \end{array}\right) : M_{1,1}, M_{2,2} \in \mathrm{U}_{n/2} \text{ and } \det(M_{1,1}M_{2,2}) = 1 \right\}.$$

See the full version for the proof [10].

A simple corollary, using the fact that the Lie algebra of the unitary group is the algebra of skew-Hermitian matrices, is the following.

▶ **Corollary 3.8.** *The Lie algebra of $\mathrm{U}_n \cap G$ is*

$$\mathrm{Lie}(\mathrm{U}_n \cap G) = \left\{ \left(\begin{array}{c|c} A_0 & 0 \\ \hline 0 & A_1 \end{array}\right) \mid A_0^* = -A_0,\ A_1^* = -A_1,\ \mathrm{tr}(A_0) + \mathrm{tr}(A_1) = 0 \right\}.$$

In particular, for any such matrix A and for all sufficiently small $\varepsilon > 0$, the exponential $\exp(\varepsilon A)$ is in $\mathrm{U}_n \cap G$.

Now our subsets X and Z are contained in subgroups $D^{-1}\mathrm{U}_n D$ and $D\mathrm{U}_n D^{-1}$, respectively, so the invariant polynomials of Lemma 3.5 are invariant under left multiplication by X and right multiplication by Z as before.

Step 2: A key property of G as the containing group, given that the invariant functions of Lemma 3.5 depend on the complex conjugate of matrix entries in addition to the matrix entries themselves, is described in the following straightforward lemma:

▶ **Lemma 3.9.** *For any $M \in G$,*

$$\overline{M[i,j]} = (-1)^{i+j} \det((QMQ)_{-i,-j})$$

where $(QMQ)_{-i,-j}$ denotes the submatrix obtained by deleting the i-th row and j-th column.

▶ **Theorem 3.10.** *Let D be as in Lemma 3.6 and Y_q as above. For $A, B \in Y_q$, if we set $M = \exp(\varepsilon A)\exp(\varepsilon B)^{-1}$, then*

$$\frac{\operatorname{tr}((D^*D)^2) - \operatorname{tr}(D^*M^*D^*DMD)}{\varepsilon^2} = c + O(\varepsilon),$$

where $2(n!)^2 c$ is a nonnegative integer satisfying $c = O(q)$ and for which $c = 0$ iff $A = B$.

For a proof, see the full version of the paper [10].

We note that $|Y_q| \geq q^{n^2/4 - n/2}$ and so if we define the polynomial $r(z)$ to be 1 on 0 and 0 on any other positive integer multiple of $1/(2(n!)^2)$ up to $O(q)$, then the function

$$p_0(M) = r((\operatorname{tr}((D^*D)^2) - \operatorname{tr}(D^*M^*D^*DMD))/\varepsilon^2)$$

is a separating function in the ring of invariant functions, of the desired degree $O(q)$. The fact that this is a *polynomial* follows from Lemma 3.9. Because $\operatorname{tr}(D^*M^*D^*DMD)$ is the polynomial $p_1(M)$ of Lemma 3.5, it is invariant under the left/right action of $D^{-1}U_nD$ and DU_nD^{-1}. From step 1 we know that $X \subseteq D^{-1}U_nD$ and $Z \subseteq DU_nD^{-1}$, and hence $p_0 \in \operatorname{Inv}_{X,Z}$.

Step 3: We now turn to describing the functions f_X, f_Z, p_X, and p_Z, beginning with showing that X and Z satisfy a version of the double product property.

▶ **Lemma 3.11.** *For $x \in D^{-1}SD$ and $y \in DSD^{-1}$, $x + y = 0$ iff $x = y = 0$.*

For a proof, see the full version of the paper [10].

▶ **Lemma 3.12.** *For X, Z as above, there exist functions f_X, f_Z and polynomials p_X, p_Z over $\mathbb{C}$ satisfying the first condition of Lemma 2.11 with $d_X = d_Z = n^2/4 - n/2$.*

For a proof, see the full version of the paper [10].

This completes the proof of Theorem 3.1.

4 Conclusions and open problems

In this paper we gave an extension of the group–theoretic framework of [14] to infinite groups (Theorem 2.2). Within this framework we explored constructions in Lie groups, raising the key question: do there exist three subsets in GL_n satisfying the TPP, of size at least $q^{n^2/2 - o_n(n)}$, and admitting separating polynomials of degree at most $q^{1+o_q(1)}$? If the answer is yes, then $\omega = 2$ (Corollary 2.8).

Towards obtaining such a construction, we developed tools using invariant theory and Lie algebras to simplify the task of designing separating polynomials (Subsection 2.4). We then put these tools to use in Section 3 to obtain a construction in U_n satisfying the target degree bound, with sets of size q^m with m approaching half the ambient dimension.

This raises several directions for future research:

- Can one obtain a construction with sets of size q^m for m approaching half the ambient dimension and separating polynomials of degree at most $q^{1+o_q(q)}$, but in GL_n rather than U_n?

- Can one obtain a construction with separating polynomials of degree at most $q^{1+o_q(q)}$ in $G_n = \mathrm{GL}_n$ or $G_n = \mathrm{U}_n$ with sets of size $q^{\dim G_n/2-o(n)}$, rather than $q^{\dim G_n/2-\Theta(n)}$?
- Our general framework opens up the possibility of using other infinite groups (not necessarily Lie groups), which remains to be explored.
- Another type of construction that is now possible is one in a single, fixed infinite group (say, GL_3), with growing families of sets (X_q, Y_q, Z_q). In contrast, our current constructions require us to take n growing as well, or more generally, a growing family of containing groups.

References

1 Josh Alman and Virginia Vassilevska Williams. Limits on all known (and some unknown) approaches to matrix multiplication. *SIAM Journal on Computing*, 52(6):FOCS18-285–FOCS18-315, 2023. doi:10.1137/19M124695X.

2 Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 522–539. SIAM, 2021. doi:10.1137/1.9781611976465.32.

3 Andris Ambainis, Yuval Filmus, and François Le Gall. Fast matrix multiplication: Limitations of the Coppersmith–Winograd method. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing (STOC 2015)*, pages 585–593. ACM, 2015. doi:10.1145/2746539.2746554.

4 Andrew Baker. *Matrix groups: an introduction to Lie group theory*. Springer Undergraduate Mathematics Series. Springer-Verlag London, Ltd., London, 2002. doi:10.1007/978-1-4471-0183-3.

5 D. Bini. Relations between exact and approximate bilinear algorithms. Applications. *Calcolo*, 17(1):87–97, 1980. doi:10.1007/BF02575865.

6 Dario Bini, Milvio Capovani, Francesco Romani, and Grazia Lotti. $O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication. *Information Processing Letters*, 8(5):234–235, 1979. doi:10.1016/0020-0190(79)90113-3.

7 Jonah Blasiak, Thomas Church, Henry Cohn, Joshua A. Grochow, Eric Naslund, William F. Sawin, and Chris Umans. On cap sets and the group-theoretic approach to matrix multiplication. *Discrete Analysis*, 2017:3, 2017. doi:10.19086/da.1245.

8 Jonah Blasiak, Thomas Church, Henry Cohn, Joshua A. Grochow, and Chris Umans. Which groups are amenable to proving exponent two for matrix multiplication?, 2017. Preprint. arXiv:1712.02302.

9 Jonah Blasiak, Henry Cohn, Joshua A. Grochow, Kevin Pratt, and Chris Umans. Matrix multiplication via matrix groups. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ITCS.2023.19.

10 Jonah Blasiak, Henry Cohn, Joshua A. Grochow, Kevin Pratt, and Chris Umans. Finite matrix multiplication algorithms from infinite groups, 2024. Preprint. arXiv:2410.14905.

11 Matthias Christandl, François Le Gall, Vladimir Lysikov, and Jeroen Zuiddam. Barriers for rectangular matrix multiplication, 2020. Preprint. arXiv:2003.03019.

12 Matthias Christandl, Péter Vrana, and Jeroen Zuiddam. Barriers for fast matrix multiplication from irreversibility. *Theory of Computing*, 17:Paper No. 2, 32, 2021. doi:10.4086/toc.2021.v017a002.

13 Henry Cohn, Robert Kleinberg, Balázs Szegedy, and Christopher Umans. Group-theoretic algorithms for matrix multiplication. In *Proceedings of the 46th Annual Symposium on Foundations of Computer Science (FOCS 2005)*, pages 379–388. IEEE Computer Society, 2005. doi:10.1109/SFCS.2005.39.

14 Henry Cohn and Christopher Umans. A group-theoretic approach to fast matrix multiplication. In *Proceedings of the 44th Annual Symposium on Foundations of Computer Science (FOCS 2003)*, pages 438–449. IEEE Computer Society, 2003. doi:10.1109/SFCS.2003.1238217.

15 Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251–280, 1990. doi:10.1016/S0747-7171(08)80013-2.

16 A. M. Davie and A. J. Stothers. Improved bound for complexity of matrix multiplication. *Proceedings of the Royal Society of Edinburgh Section A: Mathematics*, 143(2):351–369, 2013. doi:10.1017/S0308210511001648.

17 Igor Dolgachev. *Lectures on invariant theory*, volume 296 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 2003. doi:10.1017/CBO9780511615436.

18 Ran Duan, Hongxun Wu, and Renfei Zhou. Faster matrix multiplication via asymmetric hashing. In *Proceedings of the 64th Annual Symposium on Foundations of Computer Science (FOCS 2023)*, pages 2129–2138. IEEE Computer Society, 2023. doi:10.1109/FOCS57990.2023.00130.

19 William Fulton and Joe Harris. *Representation theory: a first course*, volume 129 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1991. doi:10.1007/978-1-4612-0979-9.

20 Roe Goodman and Nolan R. Wallach. *Symmetry, representations, and invariants*, volume 255 of *Graduate Texts in Mathematics*. Springer, Dordrecht, 2009. doi:10.1007/978-0-387-79852-3.

21 François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC 2014)*, pages 296–303. ACM, 2014. doi:10.1145/2608628.2608664.

22 Harriet Pollatsek. *Lie groups: a problem-oriented introduction via matrix groups*. MAA Textbooks. Mathematical Association of America, Washington, DC, 2009.

23 Claudio Procesi. *Lie groups: an approach through invariants and representations*. Universitext. Springer, New York, 2007. doi:10.1007/978-0-387-28929-8.

24 A. Schönhage. Partial and total matrix multiplication. *SIAM Journal on Computing*, 10(3):434–455, 1981. doi:10.1137/0210032.

25 József Solymosi and Van H. Vu. Near optimal bounds for the Erdős distinct distances problem in high dimensions. *Combinatorica*, 28(1):113–125, 2008. doi:10.1007/s00493-008-2099-1.

26 Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4):354–356, 1969. doi:10.1007/BF02165411.

27 Volker Strassen. The asymptotic spectrum of tensors. *Journal für die Reine und Angewandte Mathematik*, 384:102–152, 1988. doi:10.1515/crll.1988.384.102.

28 Virginia Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing (STOC 2012)*, pages 887–898. ACM, 2012. doi:10.1145/2213977.2214056.

29 Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3792–3835. SIAM, 2024. doi:10.1137/1.9781611977912.134.

Differential Privacy and Sublinear Time Are Incompatible Sometimes

Jeremiah Blocki
Purdue University, West Lafayette, IN, USA

Hendrik Fichtenberger
Google Research, Zürich, Switzerland

Elena Grigorescu
University of Waterloo, ON, Canada

Tamalika Mukherjee
Columbia University, New York, NY, USA

Abstract

Differential privacy and sublinear algorithms are both rapidly emerging algorithmic themes in times of big data analysis. Although recent works have shown the existence of differentially private sublinear algorithms for many problems including graph parameter estimation and clustering, little is known regarding hardness results on these algorithms. In this paper, we initiate the study of lower bounds for problems that aim for both differentially-private and sublinear-time algorithms. Our main result is the incompatibility of both the desiderata in the general case. In particular, we prove that a simple problem based on one-way marginals yields both a differentially-private algorithm, as well as a sublinear-time algorithm, but does not admit a "strictly" sublinear-time algorithm that is also differentially private.

2012 ACM Subject Classification Theory of computation → Theory of database privacy and security; Theory of computation → Streaming, sublinear and near linear time algorithms; Security and privacy → Privacy-preserving protocols

Keywords and phrases differential privacy, sublinear algorithms, sublinear-time algorithms, one-way marginals, lower bounds

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.19

Related Version *Full Version*: `https://arxiv.org/abs/2407.07262`

Funding *Jeremiah Blocki*: Supported in part by NSF CAREER Award CNS-2047272, and NSF Award CCF-1910659.
Elena Grigorescu: Funded in part by NSF CCF-1910659, NSF CCF-2205811, and NSF CCF-2228814.
Tamalika Mukherjee: Funded in part by NSF CCF-1910659, NSF CCF-2205811, and NSF CCF-2228814. Supported in part by Amazon through the Columbia Center of Artificial Intelligence Technology, a Google Cyber NYC Award, and DARPA under contract number W911NF2110371. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Government or DARPA.

1 Introduction

While individuals have long demanded privacy-preserving analysis and processing of their data, their adoption and enforcement by governmental and private standards, policies, and jurisdictions are now accelerating. This urgency stems, in part, from the dramatic growth in the amount of data collected, aggregated, and analyzed per individual in recent years. The sheer volume of data also poses a computational challenge as resource demands scale with data size. Thus, it is expedient to develop privacy preserving algorithms for data-analysis whose resource requirements scale sub-linearly in the size of the input dataset. Two algorithmic


© Jeremiah Blocki, Hendrik Fichtenberger, Elena Grigorescu, and Tamalika Mukherjee;
licensed under Creative Commons License CC-BY 4.0



concepts that formalize these two objectives are *differential privacy* (DP) and *sublinear algorithms*. A randomized algorithm is differentially private if its output distribution does not change significantly when we slightly modify the input dataset to add/remove an individuals data i.e., a row in the dataset. Sublinear algorithms comprise classes of algorithms that have time or space complexity that is sublinear in their input size. Previous work in the intersection of both fields has promoted several classical sublinear algorithms to differentially private sublinear time algorithms for the same problems, e.g., sublinear-time clustering [3], graph parameter estimation [4] or sublinear-space heavy hitters in streaming [27, 6].

Intuitively, one might expect that privacy and sublinear time necessarily have a symbiotic relationship, i.e., if only a fraction of the data is processed, a significant amount of sensitive information may remain unread. Recent work [5] demonstrated that if a function $f\colon D \to \mathbb{R}$ has low global sensitivity (i.e., f is amenable to DP) and there exists a *sufficiently accurate* sublinear-time approximation algorithm for f, then there exists an accurate sublinear time DP approximation for f. This lead [5] to ask whether or not a similar transformation might apply for functions $f\colon D \to \mathbb{R}^d$ with multi-dimensional output. In this paper, we provide an example of a function $f\colon D \to \mathbb{R}^d$ with the following properties (1) there is an efficient sublinear time approximation algorithm for f, (2) there is a differentially private approximation algorithm for f running in time $O(|D|)$, and (3) any accurate differentially private approximation algorithm must run in time $\tilde{\Omega}(|D|)$. Thus, the intuition that privacy and sublinear time algorithms necessarily have a symbiotic relationship is incorrect.

Existing Model for DP Lower Bounds. Consider a database $D \in \{0,1\}^{n\times d}$ with n rows where each row corresponds to an individual's record with d binary attributes. The model in this case is that the database D consists of a sample (according to a uniform or possibly adversarial distribution) from a larger population (or universe), and we are interested in answering queries on the sample (e.g., what fraction of individual records in D satisfy some property q?). One of the interesting questions in DP lower bounds is the following: Suppose we fix a family of queries $\mathcal{Q}$ and the dimensionality of database records d, then what is the sample complexity required to achieve DP and statistical accuracy for $\mathcal{Q}$? Here the *sample complexity* is defined as the minimum number of records n such that there exists a (possibly computationally unbounded) algorithm that achieves both DP and accuracy.

A key problem that has been at the center of addressing this question is the *one-way marginal problem*. The one-way marginal problem takes a database $D \in \{0,1\}^{n\times d}$ and releases the average values of all d columns. The best known private algorithm which has running time polynomial in the universe size is based on the multiplicative weights mechanism, and it achieves $(O(1), o(1/n))$-DP for $n \in O(\sqrt{d}\log|\mathcal{Q}|)$ [17]. Any pure DP algorithm for this problem requires $n \geq \Omega(d)$ samples [18], while [8] showed that $n \geq \tilde{\Omega}(\sqrt{d})$ samples are necessary to solve this problem with approximate DP and within an additive error of $1/3$.

Existing Models for Sublinear-Time Algorithms. The works on sublinear-time algorithms utilize different input models, many of them tailored to the representation of the input, e.g., whether it is a function or a graph. These models typically define query oracles, i.e., mechanisms to access the input in a structured way. For example, the dense graph model [16] defines an adjacency query oracle that, given a pair of indices (i, j), returns the entry $A(i, j)$ of the adjacency matrix A of the input graph. Query oracles enable an analysis of what parts, and more generally how much of the input was accessed by an algorithm. Since the fraction of input read is a lower bound for the time complexity of an algorithm, query access models are crucial to prove both sublinear-time upper and lower bounds.

Our Model. A challenge of proving lower bounds for sublinear time, differentially private algorithms lies in devising and applying a technique for analysis that combines the properties of both models. Lower bounds for differential privacy state a lower bound on the sample complexity that is required to guarantee privacy and non-trivial accuracy. These bounds do not state how much of the input an algorithm needs to *read* to guarantee privacy and accuracy, but only what input size *is required* to (potentially) enable such an algorithm. On the other hand, lower bounds for sublinear time algorithms state a bound on the time complexity as a function of the input size m. Note that time complexity is at least query complexity, and a lower bound on the latter immediately implies a lower bound on time complexity as well.

In our setting, we fix the number of records n, as well as the dimensionality d of the database, i.e., our problem size is $m = n \cdot d$. We define queries in our model to be *attribute queries*, i.e., querying the j-th attribute of a row i in the database D is denoted as $D(i, j)$. We emphasize that our use of the term *query* in our model is as in the sublinear-time algorithms model, and it is different from its use in conventional DP literature. Specifically, queries in DP literature refer to types of questions that the data analyst can make to the database to infer something about the population, whereas queries in the sublinear algorithms model refer to how the algorithm can access the input dataset D. In our work, we fix a problem of interest $\mathcal{P}$ on database D (e.g., the one-way marginal problem), and we consider an algorithm that solves the problem $\mathcal{P}$ on input D. Then we are interested in understanding the minimum number of (attribute) queries that an algorithm can make to solve the problem $\mathcal{P}$ and satisfy both DP and accuracy, which we call the *query complexity*.

Result of [8] does not apply in our model. For the problem of one-way marginals, we know that $n \in \tilde{\Omega}(\sqrt{d})$ [8] records are required for any algorithm to achieve both DP and accuracy. For $m = \tilde{\Omega}(d^{3/2})$, there exists a DP algorithm that can solve this problem with $\tilde{O}(m)$ queries, i.e., the algorithm can query the entire dataset and add Gaussian noise. Using Hoeffding bounds one can analyze a simple non-private algorithm with accuracy $1/3$ that has query complexity $O(d \log d)$, which is sublinear in the problem size m. However, it is not clear whether $O(m)$ queries are necessary to achieve both DP and accuracy in our model. One might be tempted to directly apply the result of [8] to say that $\tilde{\Omega}(m)$ queries are necessary, but this does not work as the results of [8] focus on sample complexity. In particular, in our model it would be possible to distribute attribute queries across all rows (making $o(d)$ attribute queries in each row) so that every row is (partially) examined but the total number of queries is still $o(m)$. In particular, a sublinear time algorithm can substantially reduce ℓ_1 and ℓ_2 sensitivity by ensuring that the maximum number of queries in each row is $o(d)$. [1]

The sample complexity lower bound of [8] uses fingerprinting codes to show that the output of an algorithm that is both DP and reasonably accurate for the one-way marginal problem can be used to reidentify an individual record, which contradicts the DP property. Intuitively, fingerprinting codes provide the guarantee that if an algorithm obtains an accurate answer after examining *at most* c rows in the database then it is possible to reidentify at

[1] Suppose for example, that $m = d^2$ so that there are d attribute columns and d rows and consider two sublinear time algorithms: Algorithm 1 examines the first $\sqrt{d}$ rows and outputs the marginals for these samples. By contrast, Algorithm 2 uses rows $i\sqrt{d}+1$ to $(i+1)\sqrt{d}$ to compute the marginals columns $i\sqrt{d}+1$ to $(i+1)\sqrt{d}$ for each $i < \sqrt{d}$. Both algorithms examine the same number of cells in the database $d\sqrt{d}$, but the ℓ_1 (resp. ℓ_2) sensitivity of the algorithms are quite different. Algorithm 1 has ℓ_1 (resp. ℓ_2) sensitivity $\sqrt{d}$ (resp. 1) while Algorithm 2 has ℓ_1 (resp. ℓ_2) sensitivity 1 (resp. $d^{-0.25}$).

least one of the corresponding users. However, if the attacker examines more than c rows, then *we cannot prove that privacy is violated as fingerprinting codes no longer provide the guarantee that we can reidentify one of the corresponding users.* In our model, an algorithm is allowed to make arbitrary attribute queries and is not restricted to querying all attributes corresponding to a fixed row, thus it is more difficult to prove a lower bound of this nature in our model. In particular, instead of sampling c rows an attacker could distribute the attribute queries across all rows (making $c = o(d)$ attribute queries in each row). The total number of cells examined is still cd, but the overall coalition has size $d \geq c$. Fingerprinting codes provide no guarantee of being able to trace a colluder since the overall coalition (number of rows in which some query was made) is larger than c. Thus, we cannot prove that privacy is violated.

Crucially, their construction relies on the algorithm being able to query the entire row (aka record) of the database and the fact that for a fixed coalition size c a fingerprinting code can trace an individual in any coalition of size $\leq c$ with high probability, as long as the individual *actively* colluded.

Our Contribution. We give the first separation between the query complexity of a non-private, sublinear-time algorithm and a DP sublinear-time algorithm (up to a log factor). We remind the reader that a lower bound on query complexity naturally implies a lower bound on time complexity as the time taken by an algorithm must be at least the number of queries made. Thus our theorem on query complexity also gives a lower bound for sublinear-time DP algorithms.

▶ **Theorem 1** (Informal Theorem). *There exists a problem $\mathcal{P}$ of size m such that*

1. *$\mathcal{P}$ can be solved privately with $O(m)$ query and time complexity.*
2. *$\mathcal{P}$ can be solved non-privately with $O(m^{2/3}\log(m)) \in o(m)$ query and time complexity.*
3. *Any algorithm that solves $\mathcal{P}$ with $(1/3, 1/3)$-accuracy and $(O(1), o(1/m))$-DP must have $\Omega(m/\log(m)) = \tilde{\Omega}(m)$ query and time complexity.*

We note that [8] implies that any accurate, DP algorithm for $\mathcal{P}$ requires $n \in \Omega(\sqrt{d})$, and we can in fact invoke the theorem for the *hardest* case and choose m so that $n = \Theta(\sqrt{d})$.[2] For full details on the definition of the problem and the formal version of the theorem, see Definition 21 and Theorem 26. Summarized in words, $\mathcal{P}$ is solvable under differential privacy, and there exists a non-private sublinear-time algorithm to solve $\mathcal{P}$, but any DP algorithm must read (almost) the entire dataset, and thus have at least (nearly) linear running time. Our techniques build upon a rich literature of using fingerprinting codes in DP lower bounds. We note that the $\log(m)$ factor in our main result (Item 3) of Theorem 1 arises from the nearly-optimal Tardos fingerprinting code used in our lower bound construction. Thus it seems unlikely that this result can be improved unless one bypasses using fingerprinting codes entirely in the DP lower bound construction.

1.1 Technical Overview

Fingerprinting code (FPC) and DP. We start the construction of our lower bound with the privacy lower bounds based on fingerprinting codes [8]. For a set of n users and a parameter $c \leq n$, an (n, d, c)-FPC consists of two algorithms $(Gen, Trace)$. The algorithm Gen on input n outputs a codebook $C \in \{0,1\}^{n\times d}$ where each row is a codeword of user $i \in [n]$ with code

[2] We call $n = \Theta(\sqrt{d})$ the hardest case because, when n becomes larger as a function of d, [1] show that subsampling can improve the privacy/accuracy trade-off of existing DP algorithms.

length $d = d(n, c)$. It guarantees that if at most c users collude to combine their codewords into a new codeword c' and the new code satisfies some (mild) *marking condition* – namely that if every colluder has the same bit b in the j-th bit of their codeword, then the j-th bit of c' must also be b – then the $Trace$ algorithm of the fingerprinting code can identify at least one colluder with high probability.[3] Bun et al. [8] used fingerprinting codes to show that the output of any accurate algorithm for the one-way marginal problem must satisfy the marking condition for sufficiently large d, and therefore, at least one row (i.e., individual) from the database is identifiable – making this algorithm not private. In more detail, given a fingerprinting code $(Gen, Trace)$, suppose a coalition of n users builds dataset $D \in \{0,1\}^{n \times d}$ where each row corresponds to a codeword of length d from the codebook Gen. For $j \in [d]$, if every user has bit b in the j-th bit of their codeword then the one-way marginal answer for that column will be b. It is shown that any algorithm that has non-trivial accuracy for answering the one-way marginals on D can be used to obtain a codeword that satisfies the marking condition. Therefore, using $Trace$ on such a codeword leads to identifying an individual in dataset D. Since an adversary is able to identify a user in D based on the answer given by the algorithm, this clearly violates DP.

Techniques of [8] do not directly apply. In our model, an algorithm only sees a subset of the entries in the entire database via attribute queries. Suppose a coalition of $c \leq n$ users belongs to a dataset $D \in \{0,1\}^{n \times d}$ where each row corresponds to a codeword of length d. As a warm-up, let us first assume that an algorithm that solves the one-way marginal problem on input $D \in \{0,1\}^{n \times d}$ always queries for entire rows and that an adversary can simulate a query oracle to the algorithm's queries, i.e., respond with rows that exactly correspond to the set of c colluders (for more details see Section 5.1). To apply a fingerprinting code argument to such an algorithm, an adversary must identify a row from this subset of c rows by examining the output of the algorithm. However, since the accuracy guarantee of the algorithm applies (only) to the *whole* dataset, we cannot make the same argument as above to conclude that the marking condition holds for the *subsample* of rows. In other words, we need to ensure that the output of an accurate algorithm that only sees a subsample of rows can also satisfy the marking condition. The techniques of [8] do not ensure such a property.

Permute Rows and Pad and Permute Columns Fingerprinting codes (PR-PPC FPC). In order to achieve the property described above, we need to ensure that any attempt of the algorithm to spoil the marking condition would contradict its accuracy guarantees. We achieve this property by padding $O(d)$ additional columns to the codebook C to obtain $C' \in \{0,1\}^{n \times d'}$, where $d' = O(d)$, so that (codebook) columns whose output could be modified to violate the marking condition and (padded) columns whose modification would violate the accuracy guarantee are indistinguishable in the subsample with good probability. Padded columns have been used to define a variant of a fingerprinting code in previous work to achieve smooth DP lower bounds [22]. In our work, we not only need a variant of FPC with padded columns, but we also need to permute the rows of the codebook (see Section 4 for the construction). This is because we need to define a sampling procedure with certain properties for the adversary to obtain a dataset on c rows from a distribution over databases of n rows, and one way to do so is permuting the rows of the codebook and outputting the first c rows (e.g., see Theorem 15 and Theorem 27).

[3] The idea of fingerprinting codes becomes colorful when imagining a publisher who distributes advance copies to press and wants to add watermarks that are robust, e.g., against pirated copies that result from averaging the copies of multiple colluders. To see that the marking assumption is a mild condition, consider that the codeword is hidden in the much larger content.

▶ Remark 2. We note that [8] used a similar padding technique to argue about obtaining error-robust codes from "weakly-robust" codes (see Lemma 6.4 in [8]). In particular, we could argue that the property that we need for our purpose is achieved by an error robust code. However, we choose to start with a weaker construction, as we do not inherently need the error robustness property.

Secret Sharing Encoding. Finally, we overcome the assumption that the algorithm queries for entire rows by applying a secret sharing scheme to the padded codebook (see Section 5.2). In particular, an adversary can encode each row $x_i \in \{0,1\}^{d'}$ with respect to a random polynomial of degree $2d' - 1$ as a share of size $2d$. The shares are defined by the d codebook values and d random values from the field. For each query of the algorithm, the adversary answers with a share from the second half. Information theory implies that the algorithm can only recover the d codebook values after querying for all d random value shares. Thus, we obtain a derivate of the one-way marginal problem that requires the algorithm to query an entire row to reveal the padded code book row. While there exist a DP algorithm (Theorem 24) and a sublinear-time algorithm (Theorem 25) for this derived problem as well, we show that there exists no sublinear-time DP algorithm (up to a log factor) that can query for arbitrary entries in the database.

2 Related Work

Fingerprinting codes were first introduced in the context of DP lower bounds by [8]. Prior to their work, traitor-tracing schemes (which can be thought of as a cryptographic analogue of information-theoretic fingerprinting codes) were used by [12, 26] to obtain computational hardness results for DP. Subsequent works have refined and generalized the connection between DP and fingerprinting codes in many ways [23, 24, 14, 7, 21, 20, 9]. The fingerprinting code techniques of proving DP lower bounds have been used in many settings including principal component analysis [15], empirical risk minimization [2], mean estimation [8, 20], regression [9], gaussian covariance estimation [21]. Recently [22] use fingerprinting codes to give smooth lower bounds for many problems including the 1-cluster problem and k-means clustering.

In the streaming model, [10] give a separation between the space complexity of differentially private algorithms and non-private algorithms – under cryptographic assumptions they show that there exists a problem that requires exponentially more space to be solved efficiently by a DP algorithm vs a non-private algorithm. By contrast, our focus is on lower bounding the running time (query complexity) of a differentially private algorithm. Our bounds do not require any cryptographic assumptions. [19] give a lower bound in the continual release model, in particular they show that there exists a problem for which any DP continual release algorithm has error $\tilde{\Omega}(T^{1/3})$ times larger than the error of a DP algorithm in the static setting where T is the length of the stream.

3 Preliminaries

We define a database $D \in \mathcal{X}^n$ to be an ordered tuple of n rows $(x_1, \ldots, x_n) \in \mathcal{X}$, where $\mathcal{X}$ is the data universe. For our purposes, we typically take $\mathcal{X} = \{0,1\}^d$. Databases D and D' are neighboring if they differ by a single row and we denote this by $D \sim D'$. In more detail, we can replace the i-th row of a database D with some fixed element of $\mathcal{X}$ to obtain dataset $D_{-i} \sim D$. Importantly both D and D_{-i} are databases of the same size.

▶ **Definition 3** (Differential Privacy [11]). *Randomized algorithm $\mathcal{A} : \mathcal{X}^n \to \mathcal{R}$ is (ε, δ)-differentially private if for every two neighboring databases $D \sim D'$ and every subset $S \subseteq \mathcal{R}$,*

$$\Pr[\mathcal{A}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{A}(D') \in S] + \delta$$

▶ **Definition 4** (Accuracy). *Randomized algorithm $\mathcal{A} : \mathcal{X}^n \to \mathbb{R}^d$ is (α, p)-accurate for problem $\mathcal{P}$ if for every $D \in \mathcal{X}^n$, with probability at least $1 - p$, the output of $\mathcal{A}$ is a vector $a \in \{0,1\}^d$ that satisfies $|a_{\mathcal{P}}(D) - a| \leq \alpha$ where $a_{\mathcal{P}}(D)$ denotes the exact solution of the problem $\mathcal{P}$ on input D.*

The following definition of fingerprinting codes is "fully-collusion-resilient". For any coalition of users S who collectively produce a string $y \in \{0,1\}^d$ as output, as long as y satisfies the *marking condition* – for all positions $1 \leq j \leq d$, if the values x_{ij} for all users i in coalition S agree with some letter $s \in \{0,1\}$, then $y_j = s$ – then the combined codeword y can be traced back to a user in the coalition. Formally, for a codebook $C \in \{0,1\}^{n \times d}$, and a coalition $S \subseteq [n]$, we define the set of feasible codewords for C_S to be

$$F(C_S) = \{c' \in \{0,1\}^d \mid \forall j \in [d], \exists i \in S, c'_j = c_{ij}\}$$

▶ **Definition 5** (Fingerprinting Codes [8]). *For any $n, d \in \mathbb{N}, \xi \in (0,1]$, a pair of algorithms $(Gen, Trace)$ is an (n, d, c)-fingerprinting code with security ξ against a coalition of size c if Gen outputs a codebook $C \in \{0,1\}^{n \times d}$ and secret state st and for every (possibly randomized) adversary $\mathcal{A}_{FP}$, and every coalition $S \subseteq [n]$ such that $|S| \leq c$, if we set $c' \leftarrow_R \mathcal{A}_{FP}(C_S)$ then*

1. $\Pr[c' \in F(C_S) \wedge Trace(c') = \perp] \leq \xi$
2. $\Pr[Trace(c') \in [n] \setminus S] \leq \xi$

where C_S contains the rows of C given by S, and the probability is taken over the coins of C, $Trace$, and $\mathcal{A}_{FP}$. The algorithms Gen and $Trace$ may share a common state denoted as st.

▶ Remark 6. Although the adversary $\mathcal{A}_{FP}$ is defined as taking the coalition of users' rows as input, we may abuse this notation and consider the entire codebook or a different input (related to the codebook) altogether. This does not change the security guarantees of the FPC against adversary $\mathcal{A}_{FP}$ because the security guarantee holds as long as the output of $\mathcal{A}_{FP}$ is a result of the users in the coalition S actively colluding.

▶ **Theorem 7** (Tardos Fingerprinting Code [25]). *For every $n \in \mathbb{N}$ and $4 \leq c \leq n$, there exists an (n, d, c)-fingerprinting code of length $d = O(c^2 \log(n/\xi))$ with security $\xi \in [0,1]$ against coalitions of size c.*

▶ **Theorem 8** (Gaussian Mechanism, [13]). *Let $\varepsilon, \delta \in (0,1)$ and $f : \mathbb{N}^d \to \mathbb{R}^d$. For $c > \sqrt{2\ln(1.25/\delta)}/\varepsilon$, the Gaussian Mechanism with standard deviation parameter $\sigma \geq c\Delta_2 f$ is (ε, δ)-DP, where Δ_2 is the ℓ_2-norm sensitivity of f.*

▶ **Lemma 9.** *For $n \geq \sqrt{200d\ln(20d)\ln(1.25/\delta)}/\varepsilon \in \tilde{\Omega}(\sqrt{d})$, given a dataset $D \in \{0,1\}^{n \times d}$, there exists a $(1/10, 1/10)$-accurate (ε, δ)-DP algorithm that solves the one-way marginals problem with $O(m)$ attribute queries, where $m = n \cdot d$.*

Proof. We note that the ℓ_2-sensitivity of the one-way marginals problem on a database $\{0,1\}^{n \times d}$ is $\sqrt{d}/n$. For $n \geq \sqrt{200d\ln(20d)\ln(1.25/\delta)}/\varepsilon \in \tilde{\Omega}(\sqrt{d})$, the Gaussian Mechanism is $(1/10, 1/10)$-accurate with

$$\sigma = \frac{\sqrt{2\ln(1.25/\delta)}}{\varepsilon} \cdot \frac{\varepsilon\sqrt{200d}}{\sqrt{2\ln(1.25/\delta) \cdot d\ln(10d)}} = \frac{1}{\sqrt{200\ln(10d)}},$$

$$\text{as } \Pr_{X \sim \mathcal{N}(0,\sigma^2)}\left[X \geq \frac{1}{10}\right] \leq 2e^{-\frac{1}{200\sigma^2}} \leq \frac{1}{20d}$$

by the Cramer-Chernoff inequality and a union bound over all d columns of the dataset. ◀

4 Permute Rows and Pad and Permute Columns Fingerprinting Codes (PR-PPC FPC)

In this section, we first introduce our pad and permute variant of the original fingerprinting codes where we Permute Rows and Pad and Permute Columns (PR-PPC (n, d, c, ℓ)-FPC). Given $(Gen, Trace)$ of an (n, d, c)-FPC, we construct Gen' and $Trace'$ in Algorithm 1 and Algorithm 2 to produce a PR-PPC (n, d, c, ℓ)-FPC. In more detail, Gen' samples codebook C and secret state st from Gen where $C \in \{0,1\}^{n \times d}$. It then permutes the rows via a random permutation π_R, after which it pads 2ℓ columns and performs another random permutation π on the columns. Then, it releases the resulting codebook $C' \in \{0,1\}^{n \times d'}$, where $d' = d + 2\ell$ and ℓ is the parameter which controls the number of columns padded to C. Note that the row permutation π_R is public while the column permutation π is part of the new secret state st'. The algorithm $Trace'$ receives an answer vector of dimension d' and uses its secret state $st' = (st, \pi)$ to feed the vector entries that correspond to the original first d columns via π^{-1} to $Trace$ and releases the output of $Trace$. We obtain the following result directly from the definition of Gen' and $Trace'$.

▶ **Corollary 10.** *Given an (n, d, c)-FPC, $\ell \geq 0$, and the corresponding PR-PPC (n, d, c, ℓ)-FPC, the properties of $Trace$ as stated by Definition 5 translate directly to $Trace'$.*

Algorithm 1 Gen'.

Require: Number of users $n \in \mathbb{N}$, number of padded 0/1 columns ℓ

1: Sample codebook $(C, st) \leftarrow Gen(n)$ such that $C \in \{0,1\}^{n \times d}$.
2: Sample random permutation $\pi : [d'] \to [d']$ where $d' := d + 2\ell$. For an $n \times d'$ matrix A, define $n \times d'$ matrix $\pi(A)$ such that $\pi(j)$-th column of $\pi(A)$ equals to the j-th column of A for every $j \in [d']$.
3: Sample random permutation $\pi_R : [n] \to [n]$. For an n-row matrix A, define $\pi_R(A)$ such that $\pi_R(i)$-th row of $\pi_R(A)$ equals to the i-th row of A for every $i \in [n]$.
4: $C^{\pi_R} \leftarrow$ Permute rows of C via random permutation π_R.
5: $C_{pad} \leftarrow$ Pad ℓ columns of all 1's and ℓ columns of all 0's to matrix C^{π_R}.
6: $C' \leftarrow$Permute the columns of C_{pad} according to random permutation π.
7: Output C' along with the new secret state $st' := (st, \pi)$ and permutation π_R.

Algorithm 2 $Trace'$.

Require: Answer vector $\mathbf{a} = (a^1, \ldots, a^{d'}) \in \{0,1\}^{d'}$, secret state $st' = (\pi, st)$

1: Output $Trace(\mathbf{a}_{og}, st)$ where $\mathbf{a}_{og} = (a^{\pi(1)}, \ldots, a^{\pi(d)}) \in \{0,1\}^d$.

We define the *feasible sample property* of an FPC below. Informally, it states that if we have an algorithm that takes a sample of rows from a codebook as input, and the algorithm's output is a feasible codeword for the entire codebook, then the same output should be a feasible codeword for the sample.

▶ **Definition 11** (Feasible Sample Property)**.** *Let $C \in \{0,1\}^{n \times d}$ be a codebook of an (n, d, c)-FPC, $S \subseteq [n]$ be a coalition and $C_S \subseteq C$ be the matrix consisting of the corresponding rows indexed by S. Given an algorithm $\mathcal{A}$ that takes as input C_S and outputs a vector $\mathbf{o} \in \{0,1\}^d$, the feasible sample property states that if $\mathbf{o} \in F(C)$, then $\mathbf{o} \in F(C_S)$.*

▶ **Lemma 12.** *PR-PPC (n, d, c, ℓ)-FPC satisfies the feasible sample property with probability at least $1 - \frac{d}{\ell}$.*

Proof. Given $(Gen', Trace')$ of PR-PPC (n, d, c, ℓ)-FPC which produces codebook $C' \in \{0,1\}^{n \times d'}$ and sampling algorithm $\mathcal{A}$ which takes as input $C'_S \subseteq C'$ and outputs a vector $\mathbf{o} \in \{0,1\}^{d'}$ where $d' = d + 2\ell$, we define the event BAD_S as $\mathbf{o} \in F(C')$ but $\mathbf{o} \notin F(C'_S)$.

We denote the indices of columns of C' in which all the bits are 1 as $C'_{|1}$ and the indices of columns in which all the bits are 0 as $C'_{|0}$. Similarly, we define $C'_{S|1}$ and $C'_{S|0}$ for the columns that are all 1 and all 0 in C'_S, respectively. Note that $C'_{|1} \subseteq C'_{S|1}$ and $C'_{|0} \subseteq C'_{S|0}$. Since by definition, algorithm $\mathcal{A}$ only has access to the set of rows in C'_S, in order for the output $\mathbf{o}$ to satisfy $\mathbf{o} \in F(C')$ but $\mathbf{o} \notin F(C'_S)$, an adversary that aims for BAD_S must flip a bit of the resulting codeword that originates from $\{C'_{S|1} \cup C'_{S|0}\} \setminus \{C'_{|1} \cup C'_{|0}\}$. In other words, the event BAD_S occurs only if the adversary identifies a column from C' that contains at least one 0 and one 1, but reduces to an all-1 or all-0 column in C'_S.

More formally, the adversary can pick the bit $b \in \{0,1\}$ resulting in a column from $C'_{S|b}$ to flip. The probability that the adversary correctly identifies a column from $C'_{S|b} \setminus C'_{|b}$ is at most $\frac{|C'_{S|b} \setminus C'_{|b}|}{|C'_{S|b}|}$. Observe that $|C'_{S|b}| \geq |C'_{|b}| \geq \ell$ due to the ℓ padded all-b columns, and therefore $|C'_{S|b} \setminus C'_{|b}| \leq |C'_{S|b}| - |C'_{|b}| \leq (d + \ell) - \ell = d$. Thus, the probability that event BAD_S occurs is at most $\max_{b \in \{0,1\}} \frac{|C'_{S|b} \setminus C'_{|b}|}{|C'_{S|b}|} \leq \frac{d}{\ell}$. ◀

5 Lower Bound

We present our lower bound for sublinear-time DP algorithms in this section. In Section 5.1 we discuss a warm-up problem, where the algorithm can only make row queries to release the one-way marginals of the dataset. We present our main result and the lower bound construction for algorithms that can make arbitrary attribute queries in Section 5.2. In the sequel, our problem space has size $m = n \cdot d = \Omega(d\sqrt{d})$ and our results will be in terms of dimension d.

5.1 Warm Up: Using a Random Oracle

In this section, we first present a warm-up problem which we call the Random Oracle Problem ($\mathcal{P}_{RO}$). This is an extension of the one-way marginals problem in the following manner – for an input dataset $D = (x_1, \ldots, x_n)$, and access to a random oracle H, the $\mathcal{P}_{RO}$ problem takes as input an encoded dataset $D_H = (z_1, \ldots, z_n)$ in which $z_i = H(i) \oplus x_i$, and outputs the one-way marginals of the underlying dataset D (see Definition 13 for a formal definition). The main intuition for introducing such a problem is that we want to force an algorithm that solves this problem to query an entire row. Recall that in our (final) model an algorithm is allowed to make arbitrary attribute queries – however, given D_H, in order to approximate or exactly compute the one-way marginals of D, an algorithm needs to query $H(i)$ for $i \in [n]$, as otherwise, by the properties of the random oracle and one-time pad (OTP), the value of x_i is information-theoretically hidden.

▶ **Definition 13** (Random Oracle Problem $\mathcal{P}_{RO}$)**.** *Given a random oracle $H : [n] \to \{0,1\}^d$, and a dataset $D = (x_1, \ldots, x_n)$ where $x_i \in \{0,1\}^d$, define dataset $D_H := (z_1, \ldots, z_n)$ where $z_i = H(i) \oplus x_i$. For simplicity of notation, we refer to the operation for obtaining D_H from D as $H(D)$. The problem $\mathcal{P}_{RO}$ on input D_H releases the one-way marginals of D.*

We use $\mathcal{P}_{RO}(D)$ to denote that $\mathcal{P}_{RO}$ releases the one-way marginals of the underlying dataset D.

Query Model. On input $D_H \in (\{0,1\}^d)^n$, an algorithm can query the random oracle H through *row queries*, i.e., given a row index $i \in [n]$ of D_H, the answer given is $H(i) \in \{0,1\}^d$. We note that our final result in this subsection will still be presented in the form of attribute queries as 1 row query translates to d attribute queries.

Observe that there exists an (ε, δ)-DP algorithm for $\mathcal{P}_{RO}(D)$ that on input D_H, queries the entire dataset via row queries to the random oracle H, i.e., it makes $dn = O(d\sqrt{d})$ queries. After obtaining the rows to the underlying dataset D it releases the one-way marginals using the Gaussian Mechanism (see Lemma 9). We also note that there exists a sublinear non-DP algorithm for $\mathcal{P}_{RO}(D)$ which makes $O(d \log d)$ queries, which is a simple corollary of Hoeffding bounds. Our goal in this section is to prove the lower bound below. Recall that $n \in \Omega(\sqrt{d})$, so the problem size is $\Omega(d\sqrt{d})$.

▶ **Theorem 14** (Lower Bound for $\mathcal{P}_{RO}$)**.** *Any algorithm that solves the problem $\mathcal{P}_{RO}$ with s attribute query complexity, $(1/3, 1/3)$-accuracy and $(O(1), o(1/s))$-DP must have $s = \Omega(d\sqrt{d/\log(d)})$.*

We present a high level overview of the proof of Theorem 14 here. We first show that given an (n, d, c)-FPC, there exists a distribution on c rows from which an adversary $\mathcal{B}$ can sample and create an n-row input instance for an algorithm $\mathcal{A}$ that accurately solves $\mathcal{P}_{RO}$ (see Theorem 15). Next we argue that the rounded output of $\mathcal{A}$, denoted as $\mathbf{a}$, is a feasible codeword for the sample of c rows as long as $\mathcal{A}$ is accurate in a non-trivial manner and $\mathbf{a}$ is feasible for the entire dataset (see Lemma 17). The adversary $\mathcal{B}$ can then use the output from $\mathcal{A}$ to (potentially) reidentify an individual from the coalition of size c. Next we relate these claims back to DP through Lemma 18, which states that if there exists a distribution $\mathcal{C}$ on $c \leq n$ row databases according to Theorem 15, then there is no (ε, δ)-DP algorithm $\mathcal{A}$ that is $(1/3, 1/3)$-accurate for $\mathcal{P}_{RO}$ with $\varepsilon = O(1)$ and $\delta = o(1/c)$. Finally, invoking the Tardos construction for fingerprinting codes in Theorem 7 gives us our lower bound.

▶ **Theorem 15.** *For every $n, d \in \mathbb{N}$, $\xi \in [0,1]$ and $c \leq n$, if there exists an (n, d, c)-fingerprinting code with security ξ, then there exists a distribution on c-row databases $\mathcal{C}_{\mathcal{S}}$, a row permutation $\pi_R : [n] \to [n]$, and an adversary $\mathcal{B}$ for every randomized algorithm $\mathcal{A}$ with row query complexity c and $(1/3, 1/3)$-accuracy for $\mathcal{P}_{RO}$ such that*

1. $\Pr_{C'_S \leftarrow \mathcal{C}_{\mathcal{S}}}[\mathcal{B}^{\mathcal{A}}(C'_S) = \perp] \leq \xi$
2. *For every $i \in [c]$, $\Pr_{C'_S \leftarrow \mathcal{C}_{\mathcal{S}}}[\mathcal{B}^{\mathcal{A}}(C'_{S_{-i}}) = \pi_R^{-1}(i)] \leq \xi$.*

The probabilities are taken over the random coins of $\mathcal{B}$ and the choice of C'_S.

Let $(Gen, Trace)$ be the promised (n, d, c)-fingerprinting code in the theorem statement. We first construct a PR-PPC (n, d, c, ℓ)-FPC with $\ell := 100d$ (see Section 4 for details).

The distribution $\mathcal{C}_{\mathcal{S}}$ on c-row databases is implicitly defined through the sampling process below

1. Let $C' \leftarrow Gen'(n, 100d)$ (see Algorithm 1) where $C' \in \{0,1\}^{n \times d'}$ and $d' = d + 100d = 101d$. Note that Gen' also outputs π_R which is a public permutation on rows.
2. Let $C'_S = (x_1, \ldots, x_c) \in \{0,1\}^{c \times d'}$ be the first c rows of $C' \in \{0,1\}^{n \times d'}$
3. Output C'_S

Next we define the privacy adversary $\mathcal{B}$.

Adversary $\mathcal{B}$ Algorithm. Adversary $\mathcal{B}$ receives C'_S as input and does the following:

1. Create a database $D = (r_1, \ldots, r_n) \in \{0,1\}^{n \times d'}$ where each row $r_i \in \{0,1\}^{d'}$ consists of 0/1 entries sampled independently and uniformly at random.

2. Given oracle access to randomized algorithm $\mathcal{A}$ which solves $\mathcal{P}_{RO}$ on input D, $\mathcal{B}$ simulates the answer to the distinct i_j-th row query (where $j \in [c]$) made by $\mathcal{A}$ to random oracle H as follows:
   a. Return $H(i_j) := r_{i_j} \oplus x_j$.
3. Let $\mathbf{a}$ be the output of $\mathcal{A}(D)$ where $\mathbf{a} \in [0,1]^{d'}$. Round each entry of $\mathbf{a}$ to $\{0,1\}$, call this new vector $\bar{\mathbf{a}} \in \{0,1\}^{d'}$.
4. Output $Trace'(\bar{\mathbf{a}})$

Analysis. We focus on proving that Property 1 and Property 2 of the theorem statement are indeed satisfied by adversary $\mathcal{B}$.

Recall the notation in Definition 13 where $\mathcal{P}_{RO}(C')$ means that $\mathcal{P}_{RO}$ releases the one-way marginals of the underlying dataset C'. We first show that $\mathcal{A}$ solving $\mathcal{P}_{RO}(H(D))$ is perfectly indistinguishable from $\mathcal{A}$ solving $\mathcal{P}_{RO}(C')$ in Lemma 16. This is necessary as $Trace'$ can only identify an individual in the coalition of size c with respect to the codebook C' produced by Gen'.

▶ **Lemma 16.** *$\mathcal{A}$ solving $\mathcal{P}_{RO}(H(D))$ is perfectly indistinguishable from $\mathcal{A}$ solving $\mathcal{P}_{RO}(C')$.*

Proof. We define the following experiments.

Real World.
1. Given $C' = (x_1, \ldots, x_n)$ where $(C', st') \leftarrow Gen'(\ell)$ with $\ell = 100d$, let $C'_S = (x_1, \ldots, x_c)$.
2. Create a database $D = (r_1, \ldots, r_n)$ where $r_i \in \{0,1\}^{d'}$ are random entries.
3. Let $\mathbf{a}$ be the output of $\mathcal{A}(D)$ where $\mathbf{a} \in [0,1]^{d'}$. Simulate H as follows:
   a. Let $i_1, \ldots, i_c$ be distinct queries made to H. For $j \in [c]$, fix $H(i_j) := r_{i_j} \oplus x_j$

Ideal World.
1. Given codebook $C' = (x_1, \ldots, x_n)$ where $(C', st') \leftarrow Gen'(\ell)$ with $\ell = 100d$, let $H(C') = (z_1, \ldots, z_n)$ (see Definition 13 for $H(\cdot)$ notation).
2. Let $\mathbf{a} \leftarrow \mathcal{A}(H(C'))$ where $\mathbf{a} \in [0,1]^{d'}$.
   a. Let $i_1, \ldots, i_c$ be distinct arbitrary queries made to H. For $j \in [c]$, H returns the following answer $H(i_j) := z_{i_j} \oplus x_j$

In the **Real World**, $\mathcal{A}$ is provided $D = (r_1, \ldots, r_n)$ as input (where D is generated in the same manner as by adversary $\mathcal{B}$), while the **Ideal World** is one in which $\mathcal{A}$ takes $H(C')$ as input. We show that $\mathcal{A}$ learns the same information in the **Real World** and the **Ideal World**, i.e., these views are perfectly indistinguishable. Observe that the only difference from the viewpoint of $\mathcal{A}$ between the **Real World** and the **Ideal World** is that H is simulated in the former via indices fixed by C'_S whereas H is queried on arbitrary indices in the latter. Since the rows of C' have already been permuted (recall Algorithm 1), by nature of the random oracle H, these two instances are perfectly indistinguishable. ◀

Recall that the security condition of the fingerprinting code (see Definition 5) only holds if $\bar{\mathbf{a}}$ is a feasible codeword for the coalition of rows, i.e., C'_S in our case. The following lemma states that if $\mathcal{A}$ is accurate for $\mathcal{P}_{RO}(C')$, then the rounded output of $\mathcal{A}$ is indeed a feasible codeword for both C' and C'_S.

▶ **Lemma 17.** *Suppose $\mathcal{A}$ is $(1/3, 1/3)$-accurate for $\mathcal{P}_{RO}(C')$. Then the rounded output $\bar{\mathbf{a}}$ from algorithm $\mathcal{A}$ is a feasible codeword for both C' and C'_S with probability at least $1 - \frac{1}{3} - \frac{1}{100}$.*

In other words, with probability at least $1 - \frac{1}{3} - \frac{1}{100}$, $\bar{\mathbf{a}} \in F(C')$ and $\bar{\mathbf{a}} \in F(C'_S)$.

Proof. Assuming that $\mathcal{A}$ is $(1/3, 1/3)$-accurate for $\mathcal{P}_{RO}(C')$, we first show that $\bar{\mathbf{a}}$ is a feasible codeword for C' with probability at least $2/3$. By the accuracy guarantee of $\mathcal{A}$, we know that for any column i_j $|\mathbf{a}_{i_j} - a_{i_j}| \leq 1/3$ where a_{i_j} is the actual 1-way marginal for column i_j with probability at least $2/3$. Thus for any column i_j of all 1's in C', $\mathbf{a}_{i_j} \geq 2/3$ which means $\bar{\mathbf{a}}_{i_j} = 1$, thus satisfying the marking condition. A similar argument holds for the case when a column is all 0's.

Next, using the fact that we use a PR-PPC $(n, d, c, 100d)$-FPC and that $\bar{\mathbf{a}} \in F(C')$ with probability at least $2/3$, we can invoke Lemma 12 which states that the feasible sample property is satisfied by our PR-PPC FPC construction. Note that in our case, the sampling algorithm described in Definition 11 is $\mathcal{A}$ together with the postprocessing step of rounding the output of $\mathcal{A}$. Also, even though $\mathcal{A}$ takes the entire dataset as input, it effectively only has access to the rows of the underlying sample via queries to $\mathcal{B}$ and thus satisfies the properties required in Definition 11. Lemma 12 states that with probability $\leq \frac{1}{100}$, $\bar{\mathbf{a}}$ is not a feasible codeword for C'_S. By a union bound we have that $1 - \frac{1}{3} - \frac{1}{100}$, $\bar{\mathbf{a}}$ must be a feasible codeword for C'_S. ◀

Proof of Theorem 15. From the above Lemma 17, we have that $\mathcal{A}$ is $(1/3, 1/3)$-accurate for $\mathcal{P}_{RO}(C')$ implies that $\bar{\mathbf{a}}$ is a feasible codeword for C'_S. By the security of the fingerprinting code, Corollary 10 and Lemma 16, we have that $\Pr[\bar{\mathbf{a}} \in F(C'_S) \wedge Trace'(\bar{\mathbf{a}}) = \perp] \leq \xi$. Since $\mathcal{B}$ releases the output of $Trace'(\bar{\mathbf{a}})$, the event $\mathcal{B}^{\mathcal{A}}(C'_S) = \perp$ is identical to $Trace'(\bar{\mathbf{a}}) = \perp$. Thus Property 1 of the theorem statement which states that the probability that $\mathcal{B}$ outputs $\perp$ is bounded by ξ follows. Property 2 follows directly from the soundness property of the fingerprinting code. ◀

▶ **Lemma 18.** *Suppose there exists a distribution on $c \leq n$ row databases $\mathcal{C}_{\mathcal{S}}$ according to Theorem 15. Then there is no (ε, δ)-DP algorithm $\mathcal{A}$ with query complexity c that is $(1/3, 1/3)$-accurate for $\mathcal{P}_{RO}$ with $\varepsilon = O(1)$ and $\delta = o(1/c)$.*

Proof. Suppose C'_S is sampled from the distribution on c-row databases $\mathcal{C}_{\mathcal{S}}$ and $\mathcal{B}$ is the adversary from Theorem 15. From the lemma statement we know that $\mathcal{A}$ is $(1/3, 1/3)$-accurate, thus using Lemma 17 and Theorem 15, we have that $\Pr[\pi_R(\mathcal{B}^{\mathcal{A}}(C'_S)) \in [c]] \geq 1 - \frac{1}{3} - \frac{1}{100} - \xi \geq \Omega(1)$. By an averaging argument, this means that there exists some $i^* \in [c]$ for which $\Pr[\pi_R(\mathcal{B}^{\mathcal{A}}(C'_S)) = i^*] \geq \Omega(1/c)$. However, if $\xi = o(1/c)$ by Property 2 in Theorem 15 we have that $\Pr[\pi_R(\mathcal{B}^{\mathcal{A}}(C'_{S_{-i^*}})) = i^*] \leq \xi = o(1/c)$.

In other words, the probability of $\mathcal{B}^{\mathcal{A}}$ outputting a fixed output i^* on neighboring input databases C'_S and $C'_{S_{-i^*}}$ is different, which violates (ε, δ)-DP for any $\varepsilon = O(1)$ and $\delta = o(1/c)$. We note here that since $\mathcal{A}$ can make at most c row queries, the DP guarantee for $\mathcal{A}$ must hold for any neighboring sample of c rows. Since $\mathcal{B}$ does some postprocessing of the output from $\mathcal{A}$, and we have shown that $\mathcal{B}$ cannot be (ε, δ)-DP, this implies that $\mathcal{A}$ cannot be (ε, δ)-DP for any $\varepsilon = O(1)$ and $\delta = o(1/c)$. ◀

5.2 Using a Secret Sharing Encoding

In this section, we remove the requirement of an algorithm querying an entire row that we enforced in the previous section. We first define the security requirement of a general encoding scheme that is sufficient to construct our DP lower bound in Definition 19. We then show that the Shamir encoding as defined in Definition 20 satisfies the security requirement (see Theorem 23). We define a problem based on this secret sharing encoding called $\mathcal{P}_{SS}$ (see Definition 21) that uses the encoding to release the one-way marginals of an underlying dataset. Finally, we show that this problem cannot have a sublinear time DP algorithm with reasonable accuracy (see Theorem 26).

▶ **Definition 19** (Security Game). *Let* $\mathtt{Exp}(\mathsf{Enc}_d, \mathcal{A}, q, d, x)$ *denote the following experiment: (1) the challenger computes* $y_0 \leftarrow \mathsf{Enc}_d(x)$ *and* $y_1 \leftarrow \mathsf{Enc}_d(0^d)$*, picks a random bit* b *and outputs* $y = y_b$*. (2)* $\mathcal{A}^y(d, q, x)$ *is given oracle access to* y *and may make up to* q *queries to the string* y*. (3) The game ends when the attacker* $\mathcal{A}$ *outputs a guess* b'*. (4) The output of the experiment is* $\mathtt{Exp}(\mathsf{Enc}_d, \mathcal{A}, q, d, x) = 1$ *if* $b' = b$ *and the attacker made at most* q *queries to* y*; otherwise the output of the experiment is* $\mathtt{Exp}(\mathcal{A}, q, d, x) = 0$*. We say that the scheme* Enc_d *is* $(q(d), d, \gamma(q, d))$*-secure if for all* $x \in \{0,1\}^d$ *and all attackers* $\mathcal{A}$ *making at most* q *queries we have*

$$\Pr[\mathtt{Exp}(\mathsf{Enc}_d, \mathcal{A}, q, d, x) = 1] \leq \frac{1}{2} + \gamma(q, d)$$

▶ **Definition 20** (Shamir Encoding). *Given a row* $x_i \in \{0,1\}^d$ *where* $i \in [n]$ *and a field* $\mathbb{F}$ *s.t.* $|\mathbb{F}| > 4d$ *let* $SS_d(x_i)$ *be the following encoding (1) pick random field elements* $\alpha_1^{(i)}, \ldots, \alpha_d^{(i)}, \alpha_{d+1}^{(i)}, \ldots, \alpha_{3d}^{(i)}$ *(distinct) and* $z_{d+1}^{(i)}, \ldots, z_{2d}^{(i)}$ *and define the polynomial* $p_i(\cdot)$ *of degree* $2d-1$ *s.t.* $p_i(\alpha_j^{(i)}) = x_j$ *and* $p_i(\alpha_{d+j}^{(i)}) = z_{d+j}^{(i)}$ *for* $j \leq d$*. (2) publish* $SS_d(x_i) = \left(\alpha_1^{(i)}, \ldots, \alpha_d^{(i)}, \{(\alpha_j^{(i)}, p_i(\alpha_j^{(i)}))\}_{j=d+1}^{3d}\right)$ *as share of* x_i*.*

▶ **Definition 21** (Secret Sharing Problem $\mathcal{P}_{SS}$). *Let dataset* $D := (x_1, \ldots, x_n) \in \{0,1\}^{n \times d}$*. Given* $D_S := (SS_d(x_1), \ldots, SS_d(x_n))$*, the goal of the secret-sharing problem* $\mathcal{P}_{SS}$ *is to release all the one-way marginals of dataset* D*.*

We use $\mathcal{P}_{SS,d}(D)$ *to denote that* $\mathcal{P}_{SS}$ *releases the one-way marginals of the underlying dataset* D *with dimension* d*.*

Query Model. On input D_S, an algorithm solving the $\mathcal{P}_{SS}$ problem can make attribute queries to obtain the underlying dataset D and release its one-way marginals. For a row $i \in [n]$, the i_j-th attribute query returns the pair of field elements $(\alpha_{j+d}^{(i)}, p(\alpha_{j+d}^{(i)}))$ of share $SS_d(x_i)$ for $1 \leq j \leq 2d$. We note that the prefix of $SS_d(x_i)$ given by $\alpha_1^{(i)}, \ldots, \alpha_d^{(i)}$ is published separately after an attribute query for the row i has been queried. In other words, the prefix does not count towards the query complexity of the algorithm.

▶ Remark 22. We remark that one can also define a different query model in which the prefix is released to the adversary whenever the i-th row is queried and our results still hold.

For completeness, we first show that the Shamir encoding SS_d defined in Definition 20 is $(q(d), d, 0)$-secure (as defined in Definition 19) where $q(d) = d$.

▶ **Theorem 23.** *The scheme* SS_d *is* $(d, d, 0)$*-secure.*

Proof. Let $x \in \{0,1\}^d$ and field $\mathbb{F}$ s.t. $|\mathbb{F}| > 4d$. Recall the secret sharing scheme $SS_d(x) = (\alpha_1, \ldots, \alpha_d, \{(\alpha_j, p(\alpha_j))\}_{j=d+1}^{3d})$ defined in Definition 20. We describe two experiments below where the Real World experiment simulates the view of the adversary and the Ideal World experiment just randomly outputs field elements. We will show that these two experiments are perfectly indistinguishable, and the security claim follows.

Real World(x).

1. Query $SS_d(x)$ for the first $q(d) = d$ pairs of coordinates and let the answers be the prefix $\alpha_1, \ldots \alpha_d$ and $\{(\alpha_{j+d}, z_{j+d})\}_{j \in [d]}$.
2. Output $\alpha_1, \ldots \alpha_d$ and $\{(\alpha_{j+d}, z_{j+d})\}_{j \in [d]}$

Ideal World(x).
1. Uniformly sample $\alpha'_1, \ldots, \alpha'_d, \alpha'_{d+1}, \ldots, \alpha'_{2d}, r_{d+1}, \ldots, r_{2d}$ from $\mathbb{F}$.
2. Output $\alpha'_1, \ldots \alpha'_d$ and $\{(\alpha'_{j+d}, r_{j+d})\}_{j\in[d]}$

Since by construction, the first d pairs of coordinates returned by SS_d and the prefix of size d correspond to $3d$ random field elements, the view of the **Real World** is therefore just the uniform distribution on $3d$ field elements and thus is identical to that of the view of the **Ideal World**. ◀

We present our main lower bound result in Theorem 26. Before we proceed, we first demonstrate the existence of a DP linear-time algorithm and non-DP sublinear-time algorithm for $\mathcal{P}_{SS}$ below.

▶ **Theorem 24.** *There exists a (ε, δ)-DP algorithm that solves the problem $\mathcal{P}_{SS}$ with $O(d\sqrt{d})$ attribute query complexity and $(1/10, 1/10)$-accuracy.*

Proof. On input D_S, the algorithm queries the entire dataset via attribute queries, i.e., it makes $dn = O(d\sqrt{d})$ queries. Given $SS(x_i) = (\alpha_1^{(i)}, \ldots, \alpha_d^{(i)}, \{(\alpha_j^{(i)}, p_i(\alpha_j^{(i)}))\}_{j=d+1}^{3d})$ for a row $i \in [n]$, the algorithm first recovers the polynomial p_i of degree $2d-1$ by doing Lagrange Interpolation over the $2d$ points given by $\{(\alpha_j^{(i)}, p_i(\alpha_j^{(i)}))\}_{j=d+1}^{3d}$. Then the original row x_i is obtained by evaluating $(p_i(\alpha_1^{(i)}), \ldots, p_i(\alpha_d^{(i)}))$. Once the original rows $x_i, \ldots, x_n$ are recovered in this manner, the algorithm can release the one-way marginals by adding Gaussian noise as detailed in Lemma 9. ◀

▶ **Theorem 25.** *There exists a sublinear-time algorithm that solves the problem $\mathcal{P}_{SS}$ with s attribute query complexity $O(d \log d)$ and $(1/10, 1/10)$-accuracy.*

Proof. The algorithm makes $O(d \log d)$ attribute queries and performs the same decoding procedure as outlined in the proof of Theorem 24 to obtain the underlying $\log(d)$ rows and computes the one-way marginals on this subset of rows. The accuracy of this algorithm is a simple corollary of Hoeffding bounds. Recall that $n \in \Omega(\sqrt{d})$, so the problem size is $\Omega(d\sqrt{d})$. ◀

▶ **Theorem 26** (Main Theorem). *Any algorithm that solves the problem $\mathcal{P}_{SS}$ with s attribute query complexity, $(1/3, 1/3)$-accuracy and $(O(1), o(1/n))$-DP must have $s = \Omega(d\sqrt{d/\log(d)})$.*

In order to prove Theorem 26, we follow a similar strategy as presented in the warm-up Section 5.1. Given an (n, d, c)-FPC, we first show how to construct a c-row distribution and an adversary $\mathcal{B}$ that can identify a user in the coalition of size c in Theorem 27.

▶ **Theorem 27.** *For every $n, d \in \mathbb{N}$, $\xi \in [0, 1]$ and $c \leq n$, if there exists an (n, d, c)-fingerprinting code with security ξ, then there exists a distribution on c-row databases $\mathcal{C}_\mathcal{S}$, a row permutation $\pi_R : [n] \to [n]$ and an adversary $\mathcal{B}$ for every randomized algorithm $\mathcal{A}$ with attribute query complexity cd' and $(1/3, 1/3)$-accuracy for $\mathcal{P}_{SS}$ such that*
1. $\Pr_{C'_S \leftarrow \mathcal{C}_\mathcal{S}}[\mathcal{B}^\mathcal{A}(C'_S) = \perp] \leq \xi$
2. *For every $i \in [c]$, $\Pr_{C'_S \leftarrow \mathcal{C}_\mathcal{S}}[\mathcal{B}^\mathcal{A}(C'_{S_{-i}}) = \pi_R^{-1}(i)] \leq \xi$.*

where $d' = 101d$ and the probability is over the random coins of $\mathcal{B}$ and the choice of C'_S.

Let $(Gen, Trace)$ be the promised (n, d, c)-fingerprinting code in the theorem statement. We first construct a PR-PPC (n, d, c, ℓ)-FPC with $\ell := 100d$ (see Section 4 for details).

The distribution $\mathcal{C}_\mathcal{S}$ on c-row databases is implicitly defined through the sampling process below:

1. Let $C' \leftarrow Gen'(n, d')$ (see Algorithm 1) where $C' \in \{0,1\}^{n \times d'}$ and $d' = 101d$.
2. Let $C'_S = (x_1, \ldots, x_c) \in \{0,1\}^{c \times d'}$ be the first c rows of $C' \in \{0,1\}^{n \times d'}$
3. Output C'_S

Next we define the privacy adversary $\mathcal{B}$.

Adversary $\mathcal{B}$ Algorithm. Let $\mathbb{F}$ be a finite field of order q' where $q' > 4d'$. Adversary $\mathcal{B}$ receives $C'_S = (x_1, \ldots, x_c)$ as input and feeds the algorithm $\mathcal{A}$ an input instance $C'_{\mathcal{B}}$ of $\mathcal{P}_{SS,d'}(C')$ by simulating answers to attribute queries made by $\mathcal{A}$ as described in Step 2a below. $\mathcal{B}$ then uses the rounded answer returned by $\mathcal{A}$ (Step 2b) to obtain an individual in the coalition by invoking $Trace'$ in Step 2c.

1. Initialize $q_i = 0$ for each row $i \in [n]$ and initialize a counter $t = 0$.
2. Simulate the oracle algorithm $\mathcal{A}$ with query access to an $(n \times 2d')$ database $C'_{\mathcal{B}}$:
   a. When $\mathcal{A}$ makes a fresh query (i, j), update $q_i = q_i + 1$ and
      - If $q_i \leq d'$, then set $b = q_i + d'$. Respond with a random pair of field elements $(\alpha_b^{(i)}, z_b^i)$. Record this tuple.
      - If $q_i = d' + 1$, then
        i. Increment t by one.
        ii. Define the entire polynomial p_i randomly, subject to the constraints that it is consistent with row x_t and the previous responses sent for row i: $p_i(\alpha_j^{(i)}) = x_{t,j}$ for $j \leq d'$ and $p_i(\alpha_b^{(i)}) = z_b^i$ for $j > d'$.
        iii. Send $\mathcal{A}$ the response $(\alpha_{j+d'}^{(i)}, p_i(\alpha_{j+d'}^{(i)}))$.
      - If $q_i > d' + 1$, then the polynomial p_i is already defined. Send the response $(\alpha_{j+d'}^{(i)}, p_i(\alpha_{j+d'}^{(i)}))$.
   b. When $\mathcal{A}$ outputs a vector $\mathbf{a} \in [0,1]^{d'}$, round its entries to $0, 1$ and call it $\bar{\mathbf{a}} \in \{0,1\}^{d'}$.
   c. Return $Trace'(\bar{\mathbf{a}})$.

We emphasize that although algorithm $\mathcal{A}$ can make attribute queries to more than c rows, the adversary $\mathcal{B}$ never defines a secret sharing polynomial for more than $t \leq c$ rows of the input C'_S.

▶ **Lemma 28.** *Suppose $\mathcal{A}$ is $(1/3, 1/3)$-accurate for $\mathcal{P}_{SS,d'}(C')$. Then the rounded output $\bar{\mathbf{a}}$ from algorithm $\mathcal{A}$ is a feasible codeword for both C' and C'_S with probability at least $1 - \frac{1}{3} - \frac{1}{100}$.*

In other words, with probability at least $1 - \frac{1}{3} - \frac{1}{100}$, $\bar{\mathbf{a}} \in F(C')$ and $\bar{\mathbf{a}} \in F(C'_S)$.

Proof. Assuming that $\mathcal{A}$ is $(1/3, 1/3)$-accurate for $\mathcal{P}_{SS,d'}(C')$, we first show that $\bar{\mathbf{a}}$ is a feasible codeword for C' with probability at least $2/3$. By the accuracy guarantee of $\mathcal{A}$, we know that for any column i_j with probability at least $2/3$, $|\mathbf{a}_{i_j} - a_{i_j}| \leq 1/3$ where a_{i_j} is the actual one-way marginal for column i_j. Thus for any column i_j of all 1's in C', $\mathbf{a}_{i_j} \geq 2/3$ which means $\bar{\mathbf{a}}_{i_j} = 1$, thus satisfying the marking condition. A similar argument holds for the case when a column is all 0's.

Next, using the fact that we use a PR-PPC $(n, d, c, 100d)$-FPC and that $\bar{\mathbf{a}} \in F(C')$ with probability at least $2/3$, we can invoke Lemma 12 which states that the feasible sample property is satisfied by our PR-PPC FPC construction. Note that in our case, the sampling algorithm described in Definition 11 is $\mathcal{A}$ together with the postprocessing step of rounding the output of $\mathcal{A}$. Also, even though $\mathcal{A}$ takes the entire dataset as input, it effectively only has access to the rows of the underlying sample via queries to $\mathcal{B}$ and thus satisfies the properties required in Definition 11. In particular, recall that the adversary maintains the invariant $t \leq c$. Lemma 12 states that with probability $\leq \frac{1}{100}$, $\bar{\mathbf{a}}$ is not a feasible codeword for C'_S. By a union bound we have that $1 - \frac{1}{3} - \frac{1}{100}$, $\bar{\mathbf{a}}$ must be a feasible codeword for C'_S. ◀

Proof of Theorem 27. From the above Lemma 28, we have that $\mathcal{A}$ is $(1/3, 1/3)$-accurate for $\mathcal{P}_{SS,d'}(C')$ implies that $\bar{\mathbf{a}}$ is a feasible codeword for C'_S. By the security of the underlying (n, d, c)-fingerprinting code and the corresponding security guarantee of the PR-PPC $(n, d, c, 100d)$-FPC given by Corollary 10, we have that $\Pr[\bar{\mathbf{a}} \in F(C'_S) \wedge Trace'(\bar{\mathbf{a}}) = \perp] \leq \xi$. Since $\mathcal{B}$ releases the output of $Trace'(\bar{\mathbf{a}})$, the event $\mathcal{B}^{\mathcal{A}}(C'_S) = \perp$ is identical to $Trace'(\bar{\mathbf{a}}) = \perp$. Thus Property 1 of the theorem statement which states that the probability that $\mathcal{B}$ outputs $\perp$ is bounded by ξ follows. Property 2 follows directly from the soundness property of the fingerprinting code. ◀

▶ **Corollary 29.** *$\mathcal{A}$ must make at least $c \cdot d'$ attribute queries to $C_{\mathcal{B}}$ to obtain c rows of C'_S where $d' = 101d$.*

Proof. Recall that Theorem 23 states that $SS_{d'}$ is $(d', d', 0)$-secure where $d' = 101d$. Thus, in order to obtain each row of C'_S, $\mathcal{A}$ must make at least d' cell queries. The statement follows from the fact that $\mathcal{A}$ queries for c rows in total. ◀

▶ **Lemma 30.** *Suppose there exists a distribution on $c \leq n$ row databases according to Theorem 27. Then there is no (ε, δ)-DP algorithm $\mathcal{A}$ with row query complexity c that is $(1/3, 1/3)$-accurate for $\mathcal{P}_{SS}$ with $\varepsilon = O(1)$ and $\delta = o(1/c)$.*

Proof. Suppose C'_S is sampled from the distribution on c-row databases $\mathcal{C}_{\mathcal{S}}$ and $\mathcal{B}$ is the adversary from Theorem 27. From the lemma statement we know that $\mathcal{A}$ is $(1/3, 1/3)$-accurate, thus using Lemma 28 and Theorem 27, we have that $\Pr[\pi_R(\mathcal{B}^{\mathcal{A}}(C'_S)) \in [c]] \geq 1 - \frac{1}{3} - \frac{1}{100} - \xi \geq \Omega(1)$. By an averaging argument, this means that there exists some $i^* \in [c]$ for which $\Pr[\pi_R(\mathcal{B}^{\mathcal{A}}(C'_S)) = i^*] \geq \Omega(1/c)$. However, if $\xi = o(1/c)$ by Property 2 in Theorem 27 we have that $\Pr[\pi_R(\mathcal{B}^{\mathcal{A}}(C'_{S_{-i^*}})) = i^*] \leq \xi = o(1/c)$.

In other words, the probability of $\mathcal{B}^{\mathcal{A}}$ outputting a fixed output i^* on neighboring input databases C'_S and $C'_{S_{-i^*}}$ is different which violates (ε, δ)-DP for any $\varepsilon = O(1)$ and $\delta = o(1/c)$. We note here that since $\mathcal{A}$ can make at most c row queries, the DP guarantee for $\mathcal{A}$ must hold for any neighboring sample of c rows. Since $\mathcal{B}$ does some postprocessing of the output from $\mathcal{A}$, and we have shown that $\mathcal{B}$ cannot be (ε, δ)-DP, this implies that $\mathcal{A}$ cannot be (ε, δ)-DP for any $\varepsilon = O(1)$ and $\delta = o(1/c)$. ◀

Proof of Theorem 26 . Recall that Lemma 30 states that if there exists a distribution $\mathcal{C}_{\mathcal{S}}$ on $c \leq n$ row databases, then there is no (ε, δ)-DP algorithm $\mathcal{A}$ that is $(1/3, 1/3)$-accurate for $\mathcal{P}_{SS}$ with $\varepsilon = O(1)$ and $\delta = o(1/c)$. From Theorem 27, such a distribution can be constructed from an (n, d, c)-fingerprinting code. Finally, invoking the Tardos construction for fingerprinting codes in Theorem 7, we get that the row query complexity must be $c = \Omega(\sqrt{d/\log(d)})$. Using Corollary 29, we know that the cell query complexity must be at least $c \cdot d' \geq \Omega(d\sqrt{d/\log(d)})$ where $d' = 101d$. ◀

References

1 Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, pages 6280–6290, 2018. URL: `https://proceedings.neurips.cc/paper/2018/hash/3b5020bb891119b9f5130f1fea9bd773-Abstract.html`.

2 Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014*, pages 464–473. IEEE Computer Society, 2014. `doi:10.1109/FOCS.2014.56`.

3 Jeremiah Blocki, Elena Grigorescu, and Tamalika Mukherjee. Differentially-private sublinear-time clustering. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 332–337. IEEE, 2021. doi:10.1109/ISIT45174.2021.9518014.

4 Jeremiah Blocki, Elena Grigorescu, and Tamalika Mukherjee. Privately estimating graph parameters in sublinear time. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.26.

5 Jeremiah Blocki, Elena Grigorescu, Tamalika Mukherjee, and Samson Zhou. How to make your approximation algorithm private: A black-box differentially-private transformation for tunable approximation algorithms of functions with low sensitivity. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2023*, volume 275 of *LIPIcs*, pages 59:1–59:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.APPROX/RANDOM.2023.59.

6 Jeremiah Blocki, Seunghoon Lee, Tamalika Mukherjee, and Samson Zhou. Differentially private L_2-heavy hitters in the sliding window model. In *The Eleventh International Conference on Learning Representations, ICLR 2023*. OpenReview.net, 2023.

7 Mark Bun, Thomas Steinke, and Jonathan R. Ullman. Make up your mind: The price of online queries in differential privacy. *J. Priv. Confidentiality*, 9(1), 2019. doi:10.29012/JPC.655.

8 Mark Bun, Jonathan R. Ullman, and Salil P. Vadhan. Fingerprinting codes and the price of approximate differential privacy. *SIAM J. Comput.*, 47(5):1888–1938, 2018. doi:10.1137/15M1033587.

9 T Tony Cai, Yichen Wang, and Linjun Zhang. The cost of privacy: Optimal rates of convergence for parameter estimation with differential privacy. *The Annals of Statistics*, 49(5):2825–2850, 2021.

10 Itai Dinur, Uri Stemmer, David P. Woodruff, and Samson Zhou. On differential privacy and adaptive data analysis with bounded space. In *Advances in Cryptology – EUROCRYPT 2023 – 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 14006 of *Lecture Notes in Computer Science*, pages 35–65. Springer, 2023. doi:10.1007/978-3-031-30620-4_2.

11 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC, Proceedings*, pages 265–284, 2006. doi:10.1007/11681878_14.

12 Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil P. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, pages 381–390. ACM, 2009. doi:10.1145/1536414.1536467.

13 Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014. doi:10.1561/0400000042.

14 Cynthia Dwork, Adam D. Smith, Thomas Steinke, Jonathan R. Ullman, and Salil P. Vadhan. Robust traceability from trace amounts. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015*, pages 650–669. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.46.

15 Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014*, pages 11–20. ACM, 2014. doi:10.1145/2591796.2591883.

16 Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998. doi:10.1145/285055.285060.

17 Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010*, pages 61–70. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.85.

18 Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010*, pages 705–714. ACM, 2010. doi:10.1145/1806689.1806786.

19 Palak Jain, Sofya Raskhodnikova, Satchit Sivakumar, and Adam Smith. The price of differential privacy under continual observation. In *International Conference on Machine Learning*, pages 14654–14678. PMLR, 2023. URL: https://proceedings.mlr.press/v202/jain23b.html.

20 Gautam Kamath, Jerry Li, Vikrant Singhal, and Jonathan R. Ullman. Privately learning high-dimensional distributions. In *Conference on Learning Theory, COLT 2019*, volume 99 of *Proceedings of Machine Learning Research*, pages 1853–1902. PMLR, 2019. URL: http://proceedings.mlr.press/v99/kamath19a.html.

21 Gautam Kamath, Argyris Mouzakis, and Vikrant Singhal. New lower bounds for private estimation and a generalized fingerprinting lemma. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, 2022.

22 Naty Peter, Eliad Tsfadia, and Jonathan R. Ullman. Smooth lower bounds for differentially private algorithms via padding-and-permuting fingerprinting codes. In *The Thirty Seventh Annual Conference on Learning Theory,*, volume 247 of *Proceedings of Machine Learning Research*, pages 4207–4239. PMLR, 2024. URL: https://proceedings.mlr.press/v247/peter24a.html.

23 Thomas Steinke and Jonathan R. Ullman. Between pure and approximate differential privacy. *J. Priv. Confidentiality*, 7(2), 2016. doi:10.29012/JPC.V7I2.648.

24 Thomas Steinke and Jonathan R. Ullman. Tight lower bounds for differentially private selection. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017*, pages 552–563. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.57.

25 Gábor Tardos. Optimal probabilistic fingerprint codes. *J. ACM*, 55(2):10:1–10:24, 2008. doi:10.1145/1346330.1346335.

26 Jonathan R. Ullman. Answering $n^{2+o(1)}$ counting queries with differential privacy is hard. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13*, pages 361–370. ACM, 2013. doi:10.1145/2488608.2488653.

27 Jalaj Upadhyay. Sublinear space private algorithms under the sliding window model. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6363–6372. PMLR, 09–15 June 2019. URL: http://proceedings.mlr.press/v97/upadhyay19a.html.

Estimating Euclidean Distance to Linearity

Andrej Bogdanov
University of Ottawa, Canada

Lorenzo Taschin
EPFL, Lausanne, Switzerland

Abstract

Given oracle access to a real-valued function on the n-dimensional Boolean cube, how many queries does it take to estimate the squared Euclidean distance to its closest linear function within ϵ? Our main result is that $O(\log^3(1/\epsilon) \cdot 1/\epsilon^2)$ queries suffice. Not only is the query complexity independent of n but it is optimal up to the polylogarithmic factor.

Our estimator evaluates f on pairs correlated by noise rates chosen to cancel out the low-degree contributions to f while leaving the linear part intact. The query complexity is optimized when the noise rates are multiples of Chebyshev nodes.

In contrast, we show that the dependence on n is unavoidable in two closely related settings. For estimation from random samples, $\Theta(\sqrt{n}/\epsilon + 1/\epsilon^2)$ samples are necessary and sufficient. For agnostically learning a linear approximation with ϵ mean-square regret under the uniform distribution, $\Omega(n/\sqrt{\epsilon})$ nonadaptively chosen queries are necessary, while $O(n/\epsilon)$ random samples are known to be sufficient (Linial, Mansour, and Nisan).

Our upper bounds apply to functions with bounded 4-norm. Our lower bounds apply even to ± 1-valued functions.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms; Theory of computation → Randomness, geometry and discrete structures; Theory of computation → Probabilistic computation

Keywords and phrases sublinear-time algorithms, statistical estimation, analysis of boolean functions, property testing, regression

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.20

Funding This work was supported by an NSERC discovery grant.

Acknowledgements We thank Gautam Prakriya for insightful discussions and the anonymous ITCS reviewers for corrections and helpful comments.

1 Introduction

Finding linear approximations is perhaps the most important problem in statistics and data science. *Linear regression* [11, Section 9.2] seeks to learn the "best" linear predictor ℓ for f from labeled data $(x, f(x))$. Owing to its wide applicability and computational advantages, approximation error is often measured as the minimum squared loss $\mathbf{E}[(f(x) - \ell(x))^2]$ under some distribution on examples.

Our main question of interest is whether *estimating* the minimum squared loss can be performed more efficiently than learning the predictor. A highly efficient estimator can be potentially used to decide whether learning a linear model is sensible at all.

We aim to highlight the conceptual distinctions between estimation and learning. We demonstrate that a complexity gap between the two problems is already exhibited by arguably the simplest distribution on examples: The uniform distribution over the Boolean cube $\{\pm 1\}^n$.

© Andrej Bogdanov and Lorenzo Taschin;
licensed under Creative Commons License CC-BY 4.0

Under the uniform distribution, the minimum squared loss is the distance between the function of interest viewed as a vector in 2^n-dimensional Euclidean space and the n-dimensional subspace spanned by the linear functions. In this setting estimating the squared distance is essentially equivalent to estimating the squared length of the projection.

The power of linear regression in machine learning applications is derived to a great extent from the distribution-free nature of the training algorithms. In contrast our results concern linear regression for a known distribution. We believe that distribution-specific regression is interesting for several reasons.

First, restrictive assumptions on the data distribution may improve performance. For example, distributions in which some features linearly approximate other features can be problematic for popular algorithms like gradient descent. The uniform distribution, as a model of pairwise independence, avoids such obstacles.

Second, distribution-specific algorithms sometimes serve as a stepping stone to distribution-free ones. In the context of property testing this strategy was successful in obtaining distribution-free algorithms for combinatorial properties like monotonicity [6] as well as algebraic ones like linearity and low-degree representability [6, 5].

Third, negative results about learning and estimation are stronger in the distribution-specific model. Distribution-free learning must succeed for all concepts in the given class *and* all distributions on examples. To understand the limitations it is natural to decouple the complexities of the concept class and of the distribution. Do learning and estimation remain hard even when the distribution is simple?

In this work we focus on information-theoretic measures of complexity. Our principal measure of performance is the *query complexity* m. The time complexity of all our algorithms is linear in mn, namely in the size of the labeled dataset. We study both algorithms that choose their queries and ones that are provided with uniformly random examples.

Our results

Our results are summarized in Table 1. Our main contribution is Theorem 1: The squared linear projection of f

$$\mathbf{W}^1[f] = \mathbf{E}[f^2] - \min_{\text{linear } \ell} \mathbf{E}[(f(x) - \ell(x))^2] \tag{1}$$

can be estimated within ϵ with $\tilde{O}(1/\epsilon^2)$ chosen queries.[1] In contrast, learning the requisite approximation and obtaining the same estimate from random samples both entail queries that grow at least as fast as some root of n.

Table 1 Query complexity of ϵ-estimating $\mathbf{W}^1[f]$ and ϵ-learning the linear part of f. All positive results assume $\|f\|_4 \leq 1$. All negative results hold even for ± 1-valued f.

	chosen queries	random samples
estimation	$O(\log^3(1/\epsilon) \cdot 1/\epsilon^2)$ (Theorem 1) $\Omega(1/\epsilon^2)$ (Theorem 8)	$O(\sqrt{n}/\epsilon + 1/\epsilon^2)$ (Theorem 18) $\Omega(\sqrt{n}/\epsilon + 1/\epsilon^2)$ (Theorem 19 + 8)
learning	$O(n/\epsilon)$ [8] $\Omega(n/\sqrt{\epsilon})$ non-adaptive (Theorem 11)	$O(n/\epsilon)$ [8] $\Omega(n/\sqrt{\epsilon})$ (Theorem 11)

[1] The query complexities of the squared projection and the squared distance to linearity are the same up to constant factor since $\mathbf{E}[f^2]$ can be estimated from $O(1/\epsilon^2)$ samples assuming $\|f\|_4$ is bounded. For similar reasons so are the squared projection and distance to the space of affine functions.

We found this result surprising from a Fourier-analytic perspective. Our quantity of interest is the sum of squares of all n first-level Fourier coefficients of f. Up to the polylogarithmic factor, estimating the sum has the same query complexity as estimating any of its individual terms, or estimating the squared mean of f.

In the decision version of estimation also known as *tolerant testing* [10], the objective is to distinguish functions that are θ-close to linear from those that are $\theta + \epsilon$-far. In particular, the algorithm of Theorem 1 is a tolerant tester for any value of θ. In the intolerant limit $\theta = 0$, algorithms with $O(1/\epsilon)$ query complexity are known [1, 3]. For functions over $\mathbb{R}^n$ the query complexity is also $O(1/\epsilon)$ under Gaussian measure [7], and $O(\log(1/\epsilon) \cdot 1/\epsilon)$ under arbitrary measure [5]. With the exception of [7], these testers appear not to be robust against small perturbations as their soundness is analyzed with respect to the Hamming distance. The tester of Khot and Moshkovitz [7] can be viewed as an estimator, albeit one with large constant bias.

Our positive results apply to functions with bounded 4-norm. Boundedness of the 2-norm is clearly insufficient: The point function $f(x) = 2^n \mathbb{1}(x = a)$ for a random point a has unit 2-norm and unit squared distance to linearity, yet is indistinguishable from the all-zero function with $o(2^n)$ queries. We did not investigate whether p-norm boundedness for $2 < p < 4$ is sufficient.

In contrast, in Theorem 11 we show that proper learning of a linear hypothesis for f that is within ϵ squared distance of optimal has non-adaptive query complexity linear in n, specifically $\Omega(n/\sqrt{\epsilon})$. Our technique also yields a $\Omega(\log n/\sqrt{\epsilon})$ general (adaptive) lower bound. On the positive side, it is known that $O(n/\epsilon)$ random samples are sufficient for learning via empirical risk mininization [8]. The $\sqrt{\epsilon}$ gap between the lower and upper bounds is reminiscent of analogous gaps in distribution-free agnostic learning of deterministic concept classes [2].

In Theorems 18 and 19 we show that the sample complexity of estimating $\mathbf{W}^1[f]$ is $\sqrt{n}/\epsilon + 1/\epsilon^2$ up to constant factor. As a consequence of Theorems 18 and 11, estimation requires fewer random samples than learning in the regime $\epsilon = \omega(n^{-2/3})$.

Techniques

Estimation from chosen queries

Algorithm 1 estimates the linear projection of f by a suitably scaled product $f(x)f(y)$ evaluated on a pair (x, y) of correlated inputs. The query complexity can be decoupled from the input length n already when the pairs (x_i, y_i) are independent, marginally uniform, and of sufficiently low correlation $\rho = \mathbf{E}\, x_i y_i$. For simplicity assume f is unbiased and ± 1-valued. By orthogonal decomposition (see (2)),

$$\mathbf{E}\, f(x)f(y) = \rho\|f^{=1}\|^2 + \rho^2\|f^{=2}\|^2 + \cdots + \rho^n\|f^{=n}\|^2,$$

where $f^{=d}$ is the degree-d part of f (see precise definition in Section 2). By Parseval's identity, $\rho^{-1}f(x)f(y)$ estimates $\mathbf{W}^1[f] = \|f^{=1}\|^2$ with bias at most $\sum_{k>1}|\rho|^{k-1}\|f^{=k}\|^2 \le |\rho|$. By choosing the bias-variance tradeoff $\rho \approx \epsilon/2$, empirically averaging this estimator yields an ϵ-approximation of $\mathbf{W}^1[f]$ with $O(1/\epsilon^4)$ queries.

To improve the query complexity we work with a more general class of distributions on pairs (x, y). Barring additional information on f, it is sensible that the distribution should place equal probability on all pairs (x, y) whose difference has fixed Hamming weight. A special class of such distibutions is mixtures of ρ-correlated pairs over some distribution $d(\rho)$ on noise rates.

The uniform mixture of ρ and $-\rho$-correlated pairs already improves the query complexity to $O(1/\epsilon^3)$. The reason is that the related Fourier decomposition

$$\mathbf{E}\ (\text{sign of correlation}) f(x)f(y) = \rho\|f^{=1}\|^2 + \rho^3\|f^{=3}\|^2 + \rho^5\|f^{=5}\|^2 + \cdots$$

cancels out all even levels, thus reducing the bias of the estimator from ρ to ρ^2.

To achieve our query complexity of $O(\log^3(1/\epsilon)1/\epsilon^2)$ we construct a mixture of noise rates $\alpha_1\rho, \ldots, \alpha_\ell\rho$ that annihilates levels 2 up to $\ell - 1$ while minimizing the contribution of levels ℓ and higher. A close to optimal error is achieved using Chebyshev interpolation. We give evidence that a $\Omega(\log^2(1/\epsilon)1/\epsilon^2)$ lower bound is inherent in this approach (Claim 6), but we do not know if it can be bypassed in general.

In Remark 7 we explain why the low complexity of Algorithm 1 owes not only to the choice of interpolation scheme, but to some serendipity in the constraints (6) on the noise rates.

The proof of Theorem 8 is based on a reduction from the problem of estimating the bias of a coin.

Learning lower bound

Theorem 11 is proved in two steps. Proposition 12 addresses the regime of constant error ϵ. It is proved by exhibiting a family of $2^{\Omega(n)}$ functions whose linear projections are $\Omega(1)$-far apart. The existence of this family is established using the probabilistic method in Lemma 13. An algorithm of sublinear query complexity is unlikely to disambiguate among members of the family and cannot be accurate.

In Proposition 15 we give a self-reduction for learning linear approximations from query complexity q and constant error to query complexity $O(q/\sqrt{\epsilon})$ and error ϵ. The reduction applies to any query model.

Estimation from samples

The random variable $f(x)f(y)\langle x, y\rangle$ with random x, y is an unbiased estimator of $\mathbf{W}^1[f]$. To improve its variance at optimal query cost we average it over all $\binom{m}{2}$ pairs of examples (Algorithm 2 and Theorem 18). For constant ϵ, the lower bound in Theorem 19 is exhibited by a sign-rounded linear function ℓ with independent Gaussian coefficients. Such a function is weakly pseudorandom against $o(\sqrt{n})$ examples yet has constant correlation with ℓ. We achieve optimal dependence on ϵ by adding Gaussian noise to ℓ.

Extensions and future work

We believe that a result analogous to Theorem 1 can be obtained for functions over $\mathbb{R}^n$ under standard Gaussian measure using related techniques. It would be interesting to investigate the general setting of product distributions under Efron-Stein decomposition. The biased Boolean cube could be a good test case.

Theorem 1 can be extended to approximate the degree-d part of f with $\log^{O(d)}(1/\epsilon) \cdot 1/\epsilon^2$ queries. Both extensions might be sensible in the context of regression, where a mix of categorical (Boolean) and numerical (Gaussian) data and higher-degree models (capturing e.g. decision trees) are more realistic.

An interesting open question is whether the polylogarithmic factor can be eliminated. We speculate that this may be possible under the stronger assumption that f is bounded in infinity-norm.

Organization

Section 2 has a brief background on Fourier analysis of Boolean functions. In Sections 3, 4, and 5 we present our results on estimation from chosen queries, learning, and estimation from samples, respectively.

2 Fourier analysis over the Boolean cube

We use standard notation from Boolean function analysis [9]. All functions are real-valued over the Boolean cube $\{\pm 1\}^n$ under uniform measure. Such functions live in Hilbert space under inner product $\mathbf{E} fg$. The p-norm of a function is $\|f\|_p = \mathbf{E}[|f|^p]^{1/p}$, with $p = 2$ usually omitted. The monomials $\{\prod_{i\in S} x_i : S \subseteq [n]\}$ form an orthonormal basis. f admits orthogonal decompositions

$$f = f^{=0} + f^{=1} + \cdots + f^{=n} = \mathbf{E} f + f^{=1} + f^{\geq 2}$$

with $f^{=j}$ (resp., $f^{\geq j}$) being the projection onto the subspace spanned by monomials of degree exactly (resp., at least) j. Our main object of interest is the linear part $f^{=1}$, especially its weight

$$\mathbf{W}^1[f] = \|f^{=1}\|^2 = \mathbf{E} f f^{=1}.$$

The orthogonal decomposition can be further refined into the complete Fourier decomposition

$$f(x) = \sum\nolimits_{S\subseteq[n]} \hat{f}(S) \prod\nolimits_{i\in S} x_i, \qquad \text{where} \qquad \hat{f}(S) = \mathbf{E} f(x) \prod\nolimits_{i\in S} x_i.$$

By orthogonality, $\mathbf{E} fg = \sum \mathbf{E} f^{=i} g^{=i} = \sum \hat{f}(S)\hat{g}(S)$ (Plancherel's formula). In particular, when $f = g$

$$\|f\|^2 = \sum \|f^{=j}\|^2 = \sum \hat{f}(S)^2. \qquad \text{(Parseval's identity)}$$

By orthogonality, the distance to linearity $\mathbf{E}[(f(x) - \ell(x))^2]$ is minimized when $\ell = f^{=1}$. The minimizer equals

$$\min_{\text{linear } \ell} \mathbf{E}[(f(x) - \ell(x))^2] = \|f\|^2 - \|f^{=1}\|^2 = \mathbf{E}[f^2] - \mathbf{W}^1[f]$$

as in (1).

A ρ-biased string in $\{\pm 1\}^n$ is one in which the coordinates are independent and ρ-biased. We use $x \cdot y$ for the pointwise product of strings x and y in $\{\pm 1\}^n$. Multiplying by a ρ-biased string and averaging the outcome has the effect of dampening the higher-degree terms:

$$\mathbf{E}_{\rho\text{-biased } e}[f(x \cdot e)] = f^{=0}(x) + \rho f^{=1}(x) + \rho^2 f^{=2}(x) + \cdots \tag{2}$$

3 Estimation from chosen queries

▶ **Theorem 1.** *There is an algorithm that makes $O(\log(\|f\|_4^2/\epsilon)^3 \cdot \|f\|_4^4/\epsilon^2)$ queries to f and outputs a value within ϵ of $\mathbf{W}^1[f]$ with probability at least $2/3$.*

The algorithm takes the empirical average of m instantiations of Algorithm 1 with parameters $m = 96\ell^3/\epsilon'^2$, $\ell = \log 1/\epsilon' + 1$, $\rho = \epsilon'^{(\ell-1)^{-1}}/6$, and $\epsilon' = \epsilon/\|f\|_4^2$. (We may assume ϵ' is an inverse power of two as this doesn't change the asymptotics.)

Algorithm 1 Estimator $\tilde{W}^1[f]$ from chosen queries.

Parameters: $0 < \rho < 1$ and $\ell \in \mathbb{N}$
Setup: Set $\alpha_1, \ldots, \alpha_\ell \in [-1, 1]$ as in (5). Calculate $d_1, \ldots, d_\ell$ as in (6).
Sample i from $\{1, \ldots, \ell\}$ with probability $|d_i|/\|d\|_1$.
Sample random x and $\alpha_i\rho$-biased e_i.
Output $\tilde{W}^1[f] = \rho^{-1}\|d\|_1 \cdot \operatorname{sign} d_i \cdot f(x)f(x \cdot e_i)$

The conditional bias of this estimator given x is $f(x)g(x)$, where

$$g = \rho^{-1} \sum d_i N_{\alpha_i \rho} f(x).$$

Here N_ρ is the noise operator $N_\rho f(x) = \mathbf{E}[f(x \cdot e)]$ for a ρ-biased e. The constants d_i, α_i are chosen so that the first Fourier level of f survives but all other levels among the first $\ell - 1$ are annihilated:

$$g^{=j} = \begin{cases} 0, & \text{if } j = 0 \text{ or } 2 \le j < \ell \\ f^{=j}, & \text{if } j = 1 \end{cases} \tag{3}$$

The theorem is proved by balancing the bias and the variance of the estimator $\tilde{W}^1$. The next two claims bound the bias and the variance respectively. Their proofs are in Section 3.1.

▷ Claim 2. $|\mathbf{E}\,\tilde{W}^1[f] - \mathbf{W}^1[f]| \le \|d\|_1 \cdot \rho^{\ell-1}\|f^{\ge \ell}\|^2$.

▷ Claim 3. $\tilde{W}^1[f]$ has variance at most $\rho^{-2} \cdot \|d\|_1^2 \cdot \|f\|_4^4$.

The quantity $\|d\|_1$ governs both. We analyze it next. Expanding g in the Fourier basis and applying (2), the constraints (3) translate to

$$\rho^{-1} \sum_{i=1}^{\ell} d_i (\alpha_i \rho)^j = \begin{cases} 0, & \text{if } j = 0 \text{ or } 2 \le j < \ell \\ 1, & \text{if } j = 1. \end{cases}$$

We seek a short solution to the linear system

$$\sum_i d_i \alpha_i^j = \begin{cases} 0, & \text{if } j = 0 \text{ or } 2 \le j < \ell \\ 1, & \text{if } j = 1. \end{cases} \tag{4}$$

in unknowns $d_1, \ldots, d_\ell$. The existence of a short solution strongly depends on the choice of evaluation points $\alpha_1, \cdots, \alpha_\ell$. The Chebyshev nodes

$$\alpha_i = \cos\Big(\pi \frac{2i - 1}{2\ell}\Big) \tag{5}$$

are close to best possible for this purpose, as will be argued in Corollary 5 and Claim 6.

▷ Claim 4. $d_i = d(\alpha_i)$, where

$$d(t) = \frac{2}{\ell} \sum_{\text{odd } j\,=\,1}^{\ell-1} (-1)^{(j-1)/2} \cdot j T_j(t), \tag{6}$$

and T_j is the j-th Chebyshev polynomial of the first kind.

Proof. The first ℓ Chebyshev polynomials $T_0, \ldots, T_{\ell-1}$ are orthogonal with respect to inner product over the Chebyshev nodes:

$$\sum_i T_j(\alpha_i)T_{j'}(\alpha_i) = \begin{cases} 0, & \text{if } j \neq j', \\ \ell/2, & \text{if } j = j' \neq 0, \\ \ell, & \text{if } j = j' = 0. \end{cases}.$$

Using the facts $T_0(t) = 1, T_1(t) = t$, and that $T_j'(0)$ is zero for even j and $(-1)^{(j-1)/2}j$ for odd j we derive the unique Chebyshev expansion

$$d(t) = \sum \hat{d}(j)T_j(t). \tag{7}$$

By orthogonality, $\hat{d}(j)$ equals $(2/\ell)\sum d(\alpha_i)T_j(\alpha_i)$, when $j \neq 0$ and half that when $j = 0$. In the latter case, the first equation in (4) says that

$$\hat{d}(0) = \frac{1}{\ell}\sum_i d(\alpha_i)T_0(\alpha_i) = \frac{1}{\ell}\sum d_i = 0.$$

Otherwise, the polynomial $T_j(t) - T_j'(0)t$ has no linear term, so

$$\sum_i d(\alpha_i)(T_j(\alpha_i) - T_j'(0)\alpha_i)$$

expands as a linear combination of $\sum_i d_i\alpha_i^k$ for $k \leq j, k \neq 1$. By (4) such linear combinations vanish. Therefore if $j \neq 1$,

$$\hat{d}(j) = \frac{2}{\ell}\sum_i d_i T_j'(0)\alpha_i = \frac{2}{\ell}T_j'(0) = \begin{cases} (2/\ell)(-1)^{(j-1)/2} \cdot j, & \text{if } j \text{ is odd} \\ 0, & \text{if } j \text{ is even.} \end{cases}$$

Plugging into (7) produces the desired formula (6). ◁

▶ **Corollary 5.** $\|d\| \leq \ell$.

Proof. By orthogonality we have the explicit formula

$$\|d\|^2 = \ell\hat{d}(0)^2 + \frac{\ell}{2}\hat{d}(1)^2 + \cdots + \frac{\ell}{2}\hat{d}(\ell-1)^2 = \frac{2}{\ell}\sum_{\text{odd } j\,=\,1}^{\ell-1} j^2,$$

which is at most twice the sum of odd integers between 1 and ℓ. Those can be matched into $\ell/2$ pairs that add up to ℓ giving the ℓ^2 upper bound. ◀

Therefore $\|d\|_1 \leq \ell^{3/2}$. This bound is within a $\sqrt{\ell}$ factor of optimal for any choice of roots:

▷ Claim 6. For any choice of $-1 \leq \alpha_1, \ldots, \alpha_\ell \leq 1$, any d that solves (4) has length $\|d\|_1 \geq \ell - 1$.

Matching this bound (up to constant factor) would result in a $\log\|f\|_4^2/\epsilon$ factor improvement in query complexity. We do not know if d as in (6) achieves it already.

Proof. Set j equal to ℓ or $\ell - 1$, whichever is odd. Take a linear combination of equations (4) weighted by the coefficients t of T_j. As T_j is bounded on $[-1, 1]$,

$$\|d\|_1 \geq \left|\sum_i d_i T_j(\alpha_i)\right| = \left|\sum_{i,k} d_i\alpha_i^k t_k\right| = |t_1| = j.$$ ◀

▶ Remark 7. By the same argument, had the right-hand side of (6) been replaced by $\mathbb{1}(j = \ell)$, there would have been no solution d of 1-norm less than $2^{\ell-1}$. Thus the low complexity of Algorithm 1 owes not only to the choice of noise parameters α_i but to the fortunate form of the right-hand side in the linear system (6).

3.1 Proof of Theorem 1

Proof of Theorem 1. By Chebyshev's inequality, the empirical average of m samples of an estimator with variance v is within $2\sqrt{v/m}$ of its mean except with probability 1/4. By Claim 2 and Claim 3, with probability at least 3/4,

$$|\tilde{W}^1[f] - \mathbf{W}^1[f]| \le \|d\|_1 \rho^{\ell-1} \|f^{\ge \ell}\|^2 + \sqrt{\frac{2\rho^{-2}\|d\|_1^2}{m}} \cdot \|f\|_4^2.$$

As $\|f^{\ge\ell}\| \le \|f\|_4$, it is sufficient that the choices of ρ, ℓ, m satisfy

$$\|d\|_1 \rho^{\ell-1} \le \frac{\epsilon'}{2} \quad \text{and} \quad \frac{2\rho^{-2}\|d\|_1^2}{m} \le \frac{\epsilon'^2}{4},$$

for $\epsilon' = \epsilon/\|f\|_4^2$. By Claim 5, $\|d\|_1 \le \sqrt{\ell} \cdot \|d\|_2 \le \ell^{3/2}$. The choice $\rho = \epsilon'^{(\ell-1)^{-1}}/6$ ensures that for $\ell \ge 2$,

$$\|d\|_1 \rho^{\ell-1} \le \frac{\ell^{3/2}}{6^{\ell-1}} \cdot \epsilon' \le \frac{1}{2}\epsilon'.$$

As $\ell = \log 1/\epsilon' + 1$,

$$\frac{2\rho^{-2}\|d\|_1^2}{m} \le \frac{24\ell^3}{m},$$

which is at most $\epsilon'^2/4$ by our choice $m = 96\ell^3/\epsilon'^2$. ◀

Proof of Claim 2. The bias of the estimator is $\mathbf{E}[fg] - \mathbf{W}^1[f] = \mathbf{E}[fg] - \|f^{=1}\|^2$. The first ℓ levels of g are given by (3), so $\mathbf{E}[f^{<\ell} g^{<\ell}] = \|f^{=1}\|^2$. As for $j \ge \ell$, by (2),

$$g^{=j} = \rho^{-1} \sum d_i (N_{\alpha_i \rho} f)^{=j} = \sum d_i (\alpha_i \rho)^j f^{=j}$$

from where

$$\begin{aligned}
|\mathbf{E}\,\tilde{W}^1[f] - \mathbf{W}^1[f]| &= |\mathbf{E}[f^{\ge\ell} g^{\ge\ell}]| \\
&\le \sum_{j\ge\ell} \rho^{j-1} \sum_i |d_i \alpha_i^j| \cdot \|f^{=j}\|^2 && \text{(by triangle inequality)} \\
&\le \rho^{\ell-1}\|d\|_1 \sum_{j\ge\ell} \|f^{=j}\|^2 && (|\alpha_i| \le 1) \\
&= \rho^{\ell-1}\|d\|_1 \|f^{\ge\ell}\|^2. && \blacktriangleleft
\end{aligned}$$

Proof of Claim 3. Operator N_ρ is contractive: $\|N_\rho g\| \le \|g\|$ for all ρ and g.

$$\begin{aligned}
\mathbf{Var}\,\tilde{W}^1[f] &\le \mathbf{E}[\tilde{W}^1[f]^2] \\
&= \rho^{-2}\|d\|_1^2\, \mathbf{E}\,\mathbf{E}[f^2 \cdot N_{\alpha_i\rho}(f^2)|i] \\
&\le \rho^{-2}\|d\|_1^2\, \mathbf{E}\sqrt{\mathbf{E}[f^4|i] \cdot \mathbf{E}[N_{\alpha_i\rho}(f^2)^2|i]} && \text{by Cauchy-Schwarz} \\
&\le \rho^{-2}\|d\|_1^2 \cdot \|f^2\| \cdot \|f^2\| && \text{by contractivity} \\
&= \rho^{-2}\|d\|_1^2 \|f\|_4^4. && \blacktriangleleft
\end{aligned}$$

3.2 Lower bound

▶ **Theorem 8.** *For every $o(1/\epsilon^2)$-query algorithm A there exists a Boolean-valued f such that $|A(f) - \mathbf{W}^1[f]| > \epsilon$ with probability at least $1/4$, as long as $\epsilon = \Omega(n^{3/2}2^{-n/2})$.*

The advantage of a distinguisher D with respect to random variable f and g is the absolute difference in the probabilities that D accepts f and g. The maximum possible distinguishing advantage is known to equal the minimum possible probability of the event $f \neq g$ under all couplings of f and g, i.e., joint distributions over (f, g) that are consistent with their marginals.

▶ **Fact 9.** *Assuming $0 \leq \beta < \alpha < 0.9$, distinguishing between independent α and β-biased coin flips with advantage at least $1/4$ requires $\Omega(1/(\alpha - \beta)^2)$ samples.*

Proof. There are several proofs for the case $\beta = 0$, see for example [13]. The general case reduces to this special case. Consider the reduction that takes the outcome of a coin flip, outputs it with probability $1 - \beta$, and outputs 1 otherwise. This reduction maps unbiased coins into β-biased ones and $(\alpha - \beta)/(1 - \beta)$-biased coins into α-biased ones. Distinguishing the latter with advantage $1/4$ therefore requires $\Omega((1 - \beta)^2/(\alpha - \beta)^2)$ samples. ◀

Proof of Theorem 8. Choose any $0.5 < \beta < \alpha < 0.9$ with $\alpha - \beta = 5\epsilon$. Let f (resp., g) be a probabilistic Boolean function in which the values $f(x)$ are independent αx_1-biased, (resp., βx_1-biased). By Claim 10 $\mathbf{W}^1[f]$ is at least $\alpha^2 - \epsilon$, and $\mathbf{W}^1[g]$ is at most $\beta^2 + \epsilon$ except with probability at most $1/4$. Therefore

$$\mathbf{W}^1[f] - \mathbf{W}^1[g] \geq \alpha^2 - \beta^2 - 2\epsilon = (\alpha - \beta)(\alpha + \beta) - 2\epsilon \geq 3\epsilon$$

by our choice of α and β. Had A been ϵ-accurate with probability $3/4$, by a union bound,

$$\begin{aligned}\Pr[|A(f) - A(g)| \leq \epsilon] \leq &\Pr[|A(f) - \mathbf{W}^1[f]| \geq \epsilon] + \Pr[|A(g) - \mathbf{W}^1[g]| \geq \epsilon] \\ &+ \Pr[|\mathbf{W}^1[f] - \mathbf{W}^1[g]| \leq 3\epsilon] \\ \leq &3/4\end{aligned}$$

so $\Pr[f \neq g] \geq \Pr[|A(f) - A(g)| > \epsilon] > 1/4$. As this holds for an arbitrary coupling of f and g, the two can be $1/4$-distinguished.

We now show this is impossible, namely f and g cannot be $1/4$-distinguished with $q = o(1/\epsilon^2)$ queries. For if they could be by some algorithm A, then the following algorithm can distinguish α and β-biased coin flips with q queries, contradicting Fact 9: Whenever A makes a new query x, flip a coin, multiply the outcome by x_1 and provide this answer. The view of A when the reduction provides α and β-biased coins is identical to interactions with f and g, respectively. ◀

▷ Claim 10. Let h be a random Boolean function whose values are independent ± 1 bits. Then $|\mathbf{W}^1[h] - \mathbf{W}^1[\mathbf{E}\, h]| \leq \epsilon$ except with probability at most $O(n^3 2^{-n}/\epsilon^2)$.

Proof. By independence, $\mathbf{Var}[\hat{h}(i)] \leq 2^{-n}$ for every i so by Chebyshev's inequality, $|\hat{h}(i) - \mathbf{E}\,\hat{h}(i)| \leq \epsilon/2n$ except with probability $O(n^2 2^{-n}/\epsilon^2)$. Therefore $|\hat{h}(i)^2 - \mathbf{E}[\hat{h}(i)]^2| = |\hat{h}(i) + \mathbf{E}\,\hat{h}(i)| \cdot |\hat{h}(i) - \mathbf{E}\,\hat{h}(i)| \leq \epsilon/n$. The claim follows from a union bound and the triangle inequality. ◁

For this argument to result in a $\Omega(1/\epsilon^2)$ lower bound it is essential that α and β be bounded away both from 0 and from 1: Had they been too close to 1 the coins would have been distinguishable from $O(1/\epsilon)$ samples. Had they been too close to 0, the typical distance between $W^1[f]$ and $W^1[g]$ would have been $O(\epsilon^2)$.

4 Learning

In the setting of *distribution-free* (agnostic) learning of a linear approximation from independent random samples, $\Omega(n/\epsilon)$ samples are necessary for properly learning a bounded n-dimensional function with respect to expected square loss error [12]. This bound can be matched by an improper learning algorithm but not by empirical risk minimization [14].

Shamir [12] points out that in general, loss minimization (output a linear hypothesis ℓ such that $\mathbf{E}[(f(x) - \ell(x))^2]$ is ϵ-close to best possible) and model approximation (find ℓ such that $\mathbf{E}[(\ell(x) - \ell^\star(x))^2] \leq \epsilon$ for the $\ell^\star$ that minimizes $\mathbf{E}[(f(x) - \ell^\star(x))^2]$) are not equivalent problems.

In the context of learning under the uniform distribution, however, the two problems are equivalent to approximating the projection $f^{\leq 1}$ onto the space of linear functions. Linial, Mansour, and Nisan [8] showed that the approximation can be calculated from $O(n/\epsilon)$ samples via empirical risk minimization.

We show a query complexity lower bound of $\Omega(n/\sqrt{\epsilon})$ for proper agnostic learning, even when the learner is given non-adaptive query access to f.

▶ **Theorem 11.** *For every non-adaptive algorithm A that makes $o(n/\sqrt{\epsilon})$ queries and outputs an affine hypothesis, there is an $f\colon \{\pm 1\}^n \to \{\pm 1\}$ such that $\|f - A(f)\|^2 \leq \|f - f^{\leq 1}\|^2 + \epsilon$ only with probability $2^{-\Omega(n)}$ assuming $\epsilon \geq 2^{-(1-o(1))n/2}$.*

Our proof technique also gives a $\Omega(\log n/\sqrt{\epsilon})$ query complexity lower bound for adaptive A.

Theorem 11 is proved in two steps. In Proposition 12 we establish it for sufficiently small but constant ϵ. In Proposition 15 we give a reduction from agnostically learning linear approximations with constant mean-square error and q queries to the same problem with mean-square error ϵ and $O(q/\sqrt{\epsilon})$ queries. The reduction is flexible with respect to the learning model, and even to the task. It can also be applied to estimation but gives a worse dependence on ϵ than what we have in Theorems 8 and 19.

4.1 Lower bound for constant error

▶ **Proposition 12.** *For every $\kappa < 1$ there exist ϵ, η such that for n sufficiently large and for every non-adaptive algorithm A that makes at most κn queries and outputs an affine hypothesis, there exists $f\colon \{\pm 1\}^n \to \{\pm 1\}$ such that $\|f - A(f)\|^2 \leq \|f - f^{\leq 1}\|^2 + \epsilon$ with probability at most $2^{-\eta n}$.*

A $\log n - O(1)$ adaptive query complexity lower bound follows from the same assumptions as adaptive algorithms that make q Boolean-valued queries can be simulated with 2^q non-adaptive queries.

The proof relies on the following lemma, which states that there is a large set of Boolean functions whose close-to-best linear approximations are mutually far apart. Any candidate learning algorithm with too few queries does not have enough information to disambiguate between these functions and is unlikely to output an approximately correct hypothesis.

▶ **Lemma 13.** *For every $\rho < 1$ there exists $\epsilon > 0$ such that for sufficiently large n, there is a set $\mathcal{P}$ of $2^{\rho n}$ pairs of functions (f, ℓ) over domain $\{-1, 1\}^n$, where f is boolean-valued, ℓ is linear and real-valued, and*

1. *For every $(f, \ell) \in \mathcal{P}$, $\|\ell - f^{\leq 1}\| \leq \sqrt{\epsilon}$.*
2. *For all distinct $(f, \ell), (f', \ell') \in \mathcal{P}$, $\|\ell - \ell'\| \geq 4\sqrt{\epsilon}$.*

By the triangle inequality:

▶ **Corollary 14.** *For every $\rho < 1$ there exists $\epsilon > 0$ and a set $\mathcal{F}$ of $2^{\rho n}$ Boolean functions such that for every $f \neq g \in \mathcal{F}$, $f^{\leq 1}$ and $g^{\leq 1}$ are at least $2\sqrt{\epsilon}$ far apart.*

By Yao's Minimax principle, to prove Proposition 12 it suffices (and is necessary) to give a probability distribution over the possible inputs f and show that any non-adaptive algorithm that makes too few queries is likely to fail on this distribution. Our distribution is uniform over the collection from Corollary 14: As $\log|\mathcal{F}|$ is larger than the query complexity, the algorithm does not have enough information to disambiguate between candidate inputs in $\mathcal{F}$ and is therefore unlikely to be correct.

Proof of Proposition 12. Choose f at random from the set of functions $\mathcal{F}$ in Corollary 14 instantiated with ρ satisfying $1 - \rho = (1 - \kappa)/2$. Let f_Q denote the restriction of f on the query set Q. Conditioned on the choice of Q, by the law of total expectation

$$\begin{aligned}
\mathbf{E}\,\frac{1}{|\{g \in \mathcal{F}\colon g_Q = f_Q\}|} &= \sum_{a \in \mathrm{Supp} f_Q} \frac{1}{|\{g \in \mathcal{F}\colon g_Q = a\}|} \cdot \Pr_{f \sim \mathcal{F}}[f_Q = a] \\
&= \sum_{a \in \mathrm{Supp} f_Q} \frac{1}{|\mathcal{F}| \Pr_{g \sim \mathcal{F}}[g_Q = a]} \cdot \Pr_{f \sim \mathcal{F}}[f_Q = a] \\
&= \frac{|\mathrm{Supp} f_Q|}{|\mathcal{F}|} \\
&\leq \frac{2^q}{|\mathcal{F}|},
\end{aligned}$$

where q is the query complexity. By our choice of parameters this is at most $2^{-(1-\kappa)n/2}$.

The left-hand side upper bounds the probability that the algorithm succeeds. For conditioned on the view of A and the choice of f, its input is equally likely to have been any function in the set $S_f = \{g \in \mathcal{F}\colon g_Q = f_Q\}$. However, the output h of $A(f)$ could be accurate for at most one function in this set: If

$$\begin{aligned}
&\|g - h\|^2 \leq \|g^{\leq 1} - g\|^2 + \epsilon \quad \text{and} \\
&\|g' - h\|^2 \leq \|g'^{\leq 1} - g'\|^2 + \epsilon
\end{aligned}$$

by Pythagoras' theorem $\|g^{\leq 1} - h\|^2, \|g'^{\leq 1} - h\|^2 \leq \epsilon$. By the triangle inequality $g^{\leq 1}$ and $g'^{\leq 1}$ are $2\sqrt{\epsilon}$-close, so it must be that $g = g'$. Therefore $A(f)$ succeeds with probability at most $1/|S_f|$. ◀

4.2 Trading accuracy for queries

We describe a reduction R that, given access to a high-accuracy learner A that requires access to many queries, learns with low accuracy but fewer queries.

When A makes a query, the reduction flips a coin with success probability $\delta = 2\sqrt{\epsilon/\epsilon_0}$. If the coin flip succeeds, the reduction forwards the query and returns the answer to A. If it fails, the reduction answers the query randomly. When A outputs its answer h, the reduction returns h/δ.

The reduction can be implemented in any query model (random samples, non-adaptive, adaptive). We show that it is effective in the context of approximately learning linear approximations.

We show correctness for Boolean-valued functions as it matches our application, but the proposition should hold more generally under any p-norm bound.

▶ **Proposition 15.** *If A makes q queries and outputs h such that $\|h - f^{\leq 1}\|^2 \leq \epsilon$ given Boolean-valued f as input, then $R^A(f)$ makes at most $q' = 4\sqrt{\epsilon/\epsilon_0}q$ queries except with probability $2^{-\Omega(q')}$, and outputs h' such that $\|h' - f^{\leq 1}\|^2 \leq \epsilon_0$ except with probability $O(n/\epsilon 2^n)$.*

Proof.

Query complexity. The number of queries is a Binomial(q, δ) random variable. By a Chernoff bound it exceeds $2\delta q$ with probability at most $2^{-\Omega(\delta q)}$.

Correctness. The input to A provided by R is indistinguishable from a function f' obtained by corrupting f by noise of rate δ, namely $f'(x) = N(x)f(x)$, where $N(x)$ are independent δ-biased bits. By the triangle inequality,

$$\|h/\delta - f^{\leq 1}\| \leq \frac{1}{\delta}\|h - f'^{\leq 1}\| + \frac{1}{\delta}\|f'^{\leq 1} - \delta f^{\leq 1}\|.$$

By our choice of δ, the first term on the right is at most $\sqrt{\epsilon_0}/2$, so it suffices to upper bound the second one by that much. In expectation, using Parseval's identity,

$$\mathbf{E}\|f'^{\leq 1} - \delta f^{\leq 1}\|^2 = \frac{(1-\delta^2)(n+1)}{2^n}, \tag{8}$$

By Markov's inequality, $\delta^{-1}\|f'^{\leq 1} - \delta f^{\leq 1}\| \leq \sqrt{\epsilon_0}/2$ except with probability $O(n/\epsilon 2^n)$. ◀

As an aside, R can be implemented in a "complexity-preserving" manner in the sense that if A is only guaranteed to work on "low-complexity" inputs f (e.g. those of small circuit complexity) then R^A works on all inputs of slightly lower complexity. The reason is that Proposition 15 (specifically (8)) only relies on the *pairwise* independence of the noise.

Proof of Theorem 11. Fix $\kappa = 1/2$ and let ϵ_0 be the corresponding ϵ from Proposition 12. Suppose A succeeds with higher probability. By Proposition 15 R^A makes $o(n) < n/2$ queries, succeeds with probability $2^{-o(n)}$, and outputs an ϵ_0-approximation, violating Proposition 12. ◀

A $\Omega(\log n/\sqrt{\epsilon})$ adaptive query lower bound follows from the same argument.

4.3 Proof of Lemma 13

▷ Claim 16. For every ϵ there exists a Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$ such that $\|f^{\leq 1} - \ell\|^2 \leq \epsilon$ for sufficiently large n, where $\ell(x) = (\gamma/\sqrt{n})(x_1 + \cdots + x_n)$, assuming $4\sqrt{6}\gamma^2 \exp(-1/4\gamma^2) \leq \epsilon$.

The proof uses the following special case of Khintchine's inequality:

▶ **Fact 17** (Khintchine's inequality). *For every linear function $\ell : \{-1,1\}^n \to \mathbb{R}$, $\|\ell\|_4^4 \leq 3\|\ell\|_2^4$.*

Proof of Claim 16. We show that a random f from a suitable distribution satisfies the desired property with nonzero probability. For each x, $f(x)$ is a random $\{-1,1\}$ outcome with bias

$$\ell(x)\mathbb{1}(|\ell(x)| \leq 1) = \begin{cases} \ell(x), & \text{if } |\ell(x)| \leq 1 \\ 0, & \text{if not.} \end{cases}$$

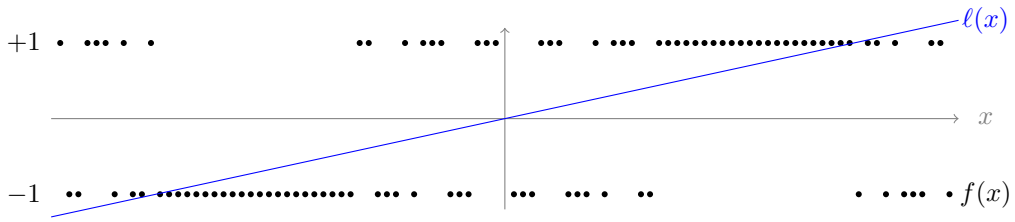


Figure 1 Illustration of f. When $|\ell(x)| \leq 1$, $f(x)$ is $\ell(x)$-biased. Otherwise, it is unbiased.

The values of f are chosen independently conditioned on $f(-x) = -f(x)$ for every x. (This folding simplifies the proof a little bit.) The construction is illustrated in Figure 1. By construction, f is balanced. By Parseval's identity,

$$\|f^{\leq 1} - \ell\|^2 = \sum_{i=1}^{n} (\hat{f}(i) - \gamma/\sqrt{n})^2. \tag{9}$$

We analyze the concentration of $\hat{f}(i)$. In expectation, and by linearity of expectation,

$$\begin{aligned}
\mathbf{E}\,\hat{f}(i) &= \mathbf{E}[x_i f(x)] \\
&= \mathbf{E}\,\mathbf{E}[x_i f(x)|x] \\
&= \mathbf{E}[x_i \ell(x) \mathbb{1}(|\ell(x)| \leq 1)] \\
&= \mathbf{E}[x_i \ell(x)] - \mathbf{E}[x_i \ell(x) \mathbb{1}(|\ell(x)| > 1)] \\
&= \frac{\gamma}{\sqrt{n}} - \hat{g}(i),
\end{aligned}$$

where g is the function $g(x) = \ell(x)\mathbb{1}(|\ell(x)| > 1)$. As g is a symmetric function, its first-level Fourier coefficients are equal and must therefore have absolute value

$$\begin{aligned}
|\hat{g}(i)| &= \sqrt{\frac{\hat{g}(1)^2 + \cdots + \hat{g}(n)^2}{n}} \\
&\leq \sqrt{\frac{\mathbf{E}[g(x)^2]}{n}} && \text{by Parseval's identity} \\
&= \frac{1}{\sqrt{n}} \sqrt{\mathbf{E}[\ell(x)^2 \mathbb{1}(|\ell(x)| > 1)]} \\
&\leq \frac{1}{\sqrt{n}} \sqrt[4]{\mathbf{E}[\ell(x)^4] \Pr[|\ell(x)| > 1]} && \text{by Cauchy-Schwarz} \\
&\leq \frac{1}{\sqrt{n}} \sqrt[4]{3\|\ell\|^4 \Pr[|\ell(x)| > 1]} && \text{by Fact 17} \\
&\leq \frac{\sqrt[4]{6}\gamma}{\sqrt{n}} \exp\left(-\frac{1}{8\gamma^2}\right) && \text{by Hoeffding's bound} \\
&\leq \sqrt{\epsilon/4n} && \text{by the assumption on } \gamma.
\end{aligned}$$

As $\hat{f}(i)$ is the average of $2^n/2$ independent outcomes, by Hoeffding's bound

$$\Pr\left[|\hat{f}(i) - \mathbf{E}\,\hat{f}(i)| > \sqrt{\epsilon/4n}\right] \leq 2\exp(-2^n\epsilon/8n).$$

Assuming n is sufficiently large so that $2\exp(-2^n\epsilon/8n)$ is less than $1/n$, by the triangle inequality and a union bound, $|\hat{f}(i) - \gamma/\sqrt{n}| \leq \sqrt{\epsilon/n}$ for all i with positive probability. By (9) there must exist a choice of f for which $\|f^{\leq 1} - \ell\|^2 \leq \epsilon$. ◀

Proof of Lemma 13. Let γ satisfy $4\sqrt{6}\gamma^2\exp(-1/4\gamma^2)=\epsilon$, $\mathcal{C}$ be a code over $\{-1,1\}^n$ of rate ρ and mininum relative Hamming distance $4\epsilon/\gamma^2$. Such codes exist for sufficiently large n by the Gilbert-Varshamov bound. Let (f,ℓ) be the pair of functions from Claim 16. The collection $\mathcal{P}$ consists of the pairs $(f_\sigma,\ell_\sigma)\colon \sigma\in\mathcal{C}$, where $g_\sigma(x)$ is the function $g(\sigma_1x_1,\dots,\sigma_nx_n)$.

As g_σ is a permutation of g, $\|f^{\le1}_\sigma-\ell^{\le1}_\sigma\|^2=\|f-\ell\|^2\le\epsilon$ proving property 1. As for property 2, by Parseval's identity, for $\sigma\neq\sigma'\in\mathcal{C}$,

$$\|\ell_\sigma-\ell_{\sigma'}\|^2=\sum\frac{\gamma^2}{n}(\sigma_i-\sigma'_i)^2=\frac{4\gamma^2}{n}\sum\mathbb{1}(\sigma_i\neq\sigma'_i)\ge16\epsilon$$

by our choice of parameters. ◀

5 Estimation from uniform samples

The queries in Algorithm 1 come in pairwise correlated pairs $(x,x\cdot e_i)$. They can be emulated from independent pairs (x,y) after reweighting by the likelihood ratio thereby preserving the bias of the estimator. However, the variance of the likelihood ratio grows exponentially in $\Theta(n\rho^2)$, leading to a comparable blowup in the estimator variance (which becomes $\approx 2^{\Omega(n\rho^2)}/\rho^2$). Such an algorithm is not query efficient as the seemingly optimal choice of $\rho\approx1/\sqrt{n}$ still fails to attain sublinear complexity.

5.1 The algorithm

Instead of adapting Algorithm 1, our pair-based sample mean estimator directly calculates the empirical mean of the unbiased estimator $A(x,y)=f(x)f(y)\langle x,y\rangle$ averaged over all pairs of samples. The reuse of samples creates correlations, but their covariances are dominated by their individual variances.

Algorithm 2 pair-based sample mean estimator.

given samples $x^1,\dots x^m\overset{i.i.d}{\sim}\{-1,1\}^n$
return $\tilde{W}^1[f]=\binom{m}{2}^{-1}\sum_{i<j}^m A(x^i,x^j)$, where $A(x,y)=f(x)f(y)\langle x,y\rangle$

The algorithm can be implemented in complexity linear in mn (the bit complexity of the samples) by evaluating the equivalent formula

$$\tilde{W}^1[f]=\binom{m}{2}^{-1}\cdot\left(\sum_i\Big(\sum\nolimits_j f(x^j)x^j_i\Big)^2-n\sum_j f(x^j)^2\right).$$

▶ **Theorem 18.** *For $m=O(\sqrt{n}\|f\|_4^2/\epsilon+\|f\|_4^4/\epsilon^2)$ independent and uniform samples, $|\tilde{W}^1[f]-\mathbf{W}^1[f]|\le\epsilon$ with probability at least $2/3$.*

Proof. When x and y are random, A and therefore $\tilde{W}^1[f]$ is an unbiased estimator of $\mathbf{W}^1[f]$:

$$\begin{aligned}\mathbf{E}\left[A(x,y)\right]&=\mathbf{E}\left[f(x)f(y)\langle x,y\rangle\right]\\&=\sum_{i=1}^n\mathbf{E}\left[f(x)x_if(y)y_i\right]\\&=\sum_{i=1}^n\mathbf{E}\left[f(x)x_i\right]^2\\&=\sum_{i=1}^n\hat{f}(i)^2=\mathbf{W}^1[f]\end{aligned}$$

The variance of $\tilde{W}^1[f]$ is the sum of the (co)variances of pairs $A(x^i, x^j)$ and $A(x^{i'}, x^{j'})$ indexed by $i < j$ and $i' < j'$ scaled by $\binom{m}{2}^{-2}$. Only intersecting pairs have nonzero contribution. There are $\binom{m}{2}$ and $2(m-2)\binom{m}{2}$ pairs of intersection size 2 and 1, respectively, resulting in

$$\mathbf{Var}\,\tilde{W}^1[f] = \frac{2}{m(m-1)}v + \frac{4(m-2)}{m(m-1)}c$$

where for independent x, y, z,

$$\begin{aligned}
v &= \mathbf{Var}\,A(x,y) \\
&\le \mathbf{E}\,A(x,y)^2 \\
&= \mathbf{E}\,f(x)^2 f(y)^2 \langle x,y\rangle^2 \\
&\le \sqrt{\mathbf{E}\,f(x)^4 f(y)^4}\sqrt{\mathbf{E}\langle x,y\rangle^4} && \text{by Cauchy-Schwarz} \\
&= \|f\|_4^4 \cdot \sqrt{3}n && \text{by Fact 17}
\end{aligned}$$

and

$$\begin{aligned}
c &= \mathbf{Cov}[A(x,y), A(x,z)] \\
&\le \mathbf{E}\,A(x,y)A(x,z) \\
&= \mathbf{E}\,\mathbf{E}[A(x,y) \mid x]^2 \\
&= \mathbf{E}\left[f(x)^2\,\mathbf{E}\left[f(y)\langle x,y\rangle \mid x\right]^2\right] \\
&\le \sqrt{\|f\|_4^4\,\mathbf{E}\,\mathbf{E}\left[f(y)\langle x,y\rangle \mid x\right]^4} && \text{by Cauchy-Schwarz} \\
&= \|f\|_4^2 \cdot \|f^{=1}\|_4^2 && \text{by Plancherel's formula} \\
&= \|f\|_4^2 \cdot \sqrt{3}\|f^{=1}\|_2^2 && \text{by Fact 17} \\
&\le \sqrt{3}\|f\|_4^4.
\end{aligned}$$

Summing up, $\mathbf{Var}\,\tilde{W}^1[f] = O(n/m^2 + 1/m) \cdot \|f\|_4^4$. The conclusion follows by applying Chebyshev's inequality to $\tilde{W}^1[f]$. ◀

5.2 Lower bound

We show that for sample-based algorithms (unlike query-based) for this problem the dependence on the dimension is inherent. More precisely, the following theorem say that Algorithm 2 is optimal in the sample model.

▶ **Theorem 19.** *For every (possibly randomized) A there exists $f\colon \{\pm1\}^n \to \{\pm1\}$ so that when given m independent examples $(x, f(x))$ with uniform x, A outputs a value within ϵ of $\mathbf{W}^1[f]$ with probability at most $1/2 + O(m\epsilon/\sqrt{n}) + O((n/\epsilon)2^{-n})$.*

The proof in fact shows that the query complexity bound in Theorem 18 is tight in all parameters, even if $\|f\|_4$ is weakened to $\|f\|_\infty$ in the statement.

To prove Theorem 19, we construct a function f that is weakly pseudorandom given $o(\sqrt{n}/\epsilon)$ examples but for which $\mathbf{W}^1[f]$ is likely to be $\Omega(\epsilon)$. In contrast, $\mathbf{W}^1[f]$ for a random function is concentrated around 2^{-n}.

The function f is the sign of

$$g_{Z,N}(x) = \sqrt{\epsilon/n} \cdot \langle x, Z\rangle + \sqrt{1-\epsilon} \cdot N(x),$$

where Z is an n-dimensional standard normal and N is a standard normal function over $\{-1,1\}^n$ (the 2^n values $N(x)$ are independent standard normals as x ranges over $\{-1,1\}^n$.)

▷ Claim 20. The statistical distance between $(x^1, O(x^1)), \ldots, (x^m, O(x^m))$ when $O = f$ and O is a random function is at most $O(m\epsilon/\sqrt{n})$.

Proof. It is sufficient to show the claim for the real-valued functions $g = g_{Z,N}$ and N, as f and a random Boolean function are obtained by taking signs of g and N, respectively, and postprocessing cannot increase statistical distance.

We will assume without loss of generality that $x^1, \ldots, x^m$ are all distinct as repetitions can only decrease statistical distance. Fixing $x^1, \ldots, x^m$, $g(x^1), \ldots, g(x^m)$ are jointly centered Gaussian with covariance matrix

$$\Sigma_{ij} = \begin{cases} (\epsilon/n)\langle x^i, x^j\rangle & \text{if } i \neq j, \\ 1, & \text{if } i = j \end{cases}$$

The covariance matrix of $N(x^1), \ldots, N(x^m)$ is the identity Id. The conditional statistical distance given $x^1, \ldots, x^m$ is at most [4]

$$O(\|\Sigma - Id\|_F) = O\left(\frac{\epsilon}{n}\sqrt{\sum\nolimits_{i\neq j}\langle x^i, x^j\rangle^2}\right).$$

By Cauchy-Schwarz, the average, unconditional statistical distance is at most

$$O\left(\frac{\epsilon}{n}\sqrt{\sum\nolimits_{i\neq j}\mathbf{E}\langle x^i, x^j\rangle^2}\right) = O\left(\frac{\epsilon}{n}\sqrt{m^2\cdot n}\right) = O(m\epsilon/\sqrt{n}). \qquad \blacktriangleleft$$

▷ Claim 21. For every z such that $n/64 \leq \|z\|^2 \leq 4n$, $\mathbf{E}[\langle X, z\rangle \operatorname{sign} g_{z,N}(X)|N] = \Omega(\sqrt{\epsilon n})$ except with probability $O(\epsilon \cdot 2^{-n})$.

Proof. The expression inside the expectation is (up to a $\sqrt{n}$ factor) of the form $t\operatorname{sign}(\sqrt{\epsilon}t + \sqrt{1-\epsilon}N)$ for $t = \langle X, z\rangle/\sqrt{n}$. For fixed t and standard normal N

$$\mathbf{E}[t\operatorname{sign}(\sqrt{\epsilon}t + \sqrt{1-\epsilon}N)] = 2|t|\Pr(|N| < \sqrt{\epsilon/(1-\epsilon)}|t|) \geq 2|t|\Pr(|N| < \sqrt{\epsilon}|t|). \tag{10}$$

As the right-hand side is always nonnegative,

$$\begin{aligned}
&\mathbf{E}[\langle X, z\rangle \operatorname{sign} g_{z,N}(X)] \\
&\geq \mathbf{E}[\langle X, z\rangle \operatorname{sign} g_{z,N}(X) | |\langle X, z\rangle| \geq \|z\|/3] \cdot \Pr(|\langle X, z\rangle| \geq \|z\|/3) \\
&\geq 2(\|z\|/3)\Pr(|N| < \sqrt{\epsilon}/24) \cdot \Pr(|\langle X, z\rangle| \geq \|z\|/3) && \text{by (10)} \\
&= \Omega(\sqrt{n} \cdot \Pr(|N| < \sqrt{\epsilon}/24 \cdot 3)) && \text{by Khinchine's inequality} \\
&= \Omega(\sqrt{\epsilon n}).
\end{aligned}$$

By independence of the values $\operatorname{sign} g_{z,N}(x)$ across x,

$$\begin{aligned}
\mathbf{Var}\,\mathbf{E}[\langle X, z\rangle \operatorname{sign} g_{z,N}(X)|N] &= \mathbf{Var}\,\frac{1}{2^n}\sum_x \langle x, z\rangle \operatorname{sign} g_{z,N}(x) \\
&= \frac{1}{2^{2n}}\sum_x \langle x, z\rangle^2\,\mathbf{Var}\operatorname{sign} g_{z,N}(x) \\
&\leq \frac{1}{2^{2n}}\sum_x \langle x, z\rangle^2 \\
&= \frac{\|z\|^2}{2^n}.
\end{aligned}$$

The claim follows from Chebyshev's inequality. $\triangleleft$

▷ Claim 22. $\mathbf{W}^1[f] \geq \Omega(\epsilon)$ except with probability $\Omega(2^{-n})$.

Proof. By the Cauchy-Schwarz inequality, for any linear function ℓ,

$$\langle f, \ell \rangle^2 = \langle f^{\leq 1}, \ell \rangle^2 \leq \|f^{\leq 1}\|^2 \cdot \|\ell\|^2 = \mathbf{W}^1[f] \cdot \|\ell\|^2.$$

The function $\ell(x) = \langle x, Z \rangle$ has squared 2-norm $\sum Z_i^2$, which is between $n/64$ and $4n$ except with probability 2^{-n}. Applying Claim 21,

$$\mathbf{W}^1[f] \geq \frac{\langle f, \ell \rangle^2}{\|\ell\|^2} = \frac{\mathbf{E}[\langle X, Z \rangle \operatorname{sign} g_{Z,N}(X) | N, Z]^2}{\mathbf{E}[\langle X, Z \rangle^2 | Z]} = \Omega(\epsilon). \quad \blacktriangleleft$$

▷ Claim 23. For a random Boolean function r, $\mathbf{W}^1[r] \leq \epsilon$ except with probability $(n/\epsilon)2^{-n}$.

Proof. By symmetry the expectation is $n2^{-n}$. The bound follows from Markov's inequality. ◁

Proof of Theorem 19. Applying Claims 22 with ϵ in the definition of f scaled up by a suitable constant factor, $\mathbf{W}^1[f] > 3\epsilon$ except with probability $O(2^{-n})$. By Claim 23, $\mathbf{W}^1[r] \leq \epsilon$ except with probability $(n/\epsilon)2^{-n}$.

Let D be the distinguisher that accepts if A's output is greater than 2ϵ and rejects if not. If A is correct with probability at least $(1 + \delta)/2$ on every input, then D accepts f with probability at least $1/2 + \delta/2 - O(2^{-n})$ while D accepts r with probability at most $1/2 - \delta/2 + (n/\epsilon)2^{-n}$. As D's distinguishing advantage cannot exceed A's, the statistical distance between A's views on inputs f and r is at least $\delta - O((n/\epsilon)2^{-n})$. By Claim 20 it is at most $O(m\epsilon/\sqrt{n})$, from where $\delta \leq O(m\epsilon/\sqrt{n}) + O((n/\epsilon)2^{-n})$. ◀

References

1. Mitali Bafna, Srikanth Srinivasan, and Madhu Sudan. Local decoding and testing of polynomials over grids. *Random Struct. Algorithms*, 57(3):658–694, 2020. doi:10.1002/RSA.20933.
2. Shai Ben-David and Ruth Urner. The sample complexity of agnostic learning under deterministic labels. In *Proceedings of The 27th Conference on Learning Theory*, volume 35, pages 527–542, 13–15 June 2014. URL: https://proceedings.mlr.press/v35/ben-david14.html.
3. Andrej Bogdanov and Gautam Prakriya. Direct Sum and Partitionability Testing over General Groups. In *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198, pages 33:1–33:19, 2021. doi:10.4230/LIPIcs.ICALP.2021.33.
4. Luc Devroye, Abbas Mehrabian, and Tommy Reddad. The total variation distance between high-dimensional gaussians with the same mean. *arXiv preprint*, 2018. arXiv:1810.08693.
5. Noah Fleming and Yuichi Yoshida. Distribution-free testing of linear functions on $\mathbb{R}^n$. *arXiv preprint*, 2019. arXiv:1909.03391.
6. Shirley Halevy and Eyal Kushilevitz. Distribution-free property-testing. *SIAM J. Comput.*, 37(4):1107–1138, 2007. doi:10.1137/050645804.
7. Subhash Khot and Dana Moshkovitz. NP-hardness of approximately solving linear equations over reals. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 413–420, 2011. doi:10.1145/1993636.1993692.
8. Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. *Journal of the ACM (JACM)*, 40(3):607–620, 1993. doi:10.1145/174130.174138.
9. Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.
10. Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *J. Comput. Syst. Sci.*, 72(6):1012–1042, 2006. doi:10.1016/j.jcss.2006.03.002.

11 Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, USA, 2014.

12 Ohad Shamir. The sample complexity of learning linear predictors with the squared loss. *J. Mach. Learn. Res.*, 16(1):3475–3486, January 2015. doi:10.5555/2789272.2912110.

13 Madhur Tulsiani. Lecture 5: Information and coding theory. Lecture Notes, Winter 2021. https://home.ttic.edu/~madhurt/courses/infotheory2021/l5.pdf.

14 Tomas Vaškevičius and Nikita Zhivotovskiy. Suboptimality of constrained least squares and improvements via non-linear predictors. *Bernoulli*, 29, February 2023. doi:10.3150/22-BEJ1465.

Adjacency Labeling Schemes for Small Classes

Édouard Bonnet
Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France

Julien Duron
Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France

John Sylvester
Department of Computer Science, University of Liverpool, UK

Viktor Zamaraev
Department of Computer Science, University of Liverpool, UK

Abstract

A graph class admits an implicit representation if, for every positive integer n, its n-vertex graphs have a $O(\log n)$-bit (adjacency) labeling scheme, i.e., their vertices can be labeled by binary strings of length $O(\log n)$ such that the presence of an edge between any pair of vertices can be deduced solely from their labels. The famous Implicit Graph Conjecture posited that every hereditary (i.e., closed under taking induced subgraphs) factorial (i.e., containing $2^{O(n \log n)}$ n-vertex graphs) class admits an implicit representation. The conjecture was recently refuted [Hatami and Hatami, FOCS '22], and does not even hold among monotone (i.e., closed under taking subgraphs) factorial classes [Bonnet et al., ICALP '24]. However, monotone small (i.e., containing at most $n!c^n$ many n-vertex graphs for some constant c) classes do admit implicit representations.

This motivates the Small Implicit Graph Conjecture: Every hereditary small class admits an $O(\log n)$-bit labeling scheme. We provide evidence supporting the Small Implicit Graph Conjecture. First, we show that every small weakly sparse (i.e., excluding some fixed bipartite complete graph as a subgraph) class has an implicit representation. This is a consequence of the following fact of independent interest proved in the paper: Every weakly sparse small class has bounded expansion (hence, in particular, bounded degeneracy). Second, we show that every hereditary small class admits an $O(\log^3 n)$-bit labeling scheme, which provides a substantial improvement of the best-known polynomial upper bound of $n^{1-\varepsilon}$ on the size of adjacency labeling schemes for such classes. This is a consequence of another fact of independent interest proved in the paper: Every small class has neighborhood complexity $O(n \log n)$.

2012 ACM Subject Classification Mathematics of computing → Combinatorics; Mathematics of computing → Graph theory

Keywords and phrases Adjacency labeling, degeneracy, weakly sparse classes, small classes, implicit graph conjecture

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.21

Related Version *Full Version*: `https://arxiv.org/abs/2409.04821` [49]

Funding This work has been supported by the Royal Society (IES\R1\231083), by the ANR projects TWIN-WIDTH (ANR-21-CE48-0014) and Digraphs (ANR-19-CE48-0013), and also the EPSRC project EP/T004878/1: Multilayer Algorithmics to Leverage Graph Structure.

Acknowledgements We are grateful to Maksim Zhukovskii for many stimulating discussions on the topic of this paper. The first author thanks Szymon Toruńczyk for introducing him to (and answering questions on) the topic of paths with low crossing number.



© Édouard Bonnet, Julien Duron, John Sylvester, and Viktor Zamaraev;
licensed under Creative Commons License CC-BY 4.0



1 Introduction

A *class* of graphs is a set of graphs which is closed under isomorphism. For a class of graphs $\mathcal{X}$ we denote by $\mathcal{X}_n$ the set of graphs in $\mathcal{X}$ with vertex set $[n]$. Let $\mathcal{X}$ be a class of graphs and $b : \mathbb{N} \to \mathbb{N}$ be a function. A $b(n)$-*bit adjacency labeling scheme* (or simply $b(n)$-*bit labeling scheme*) for $\mathcal{X}$ is a pair (encoder, decoder) of algorithms where for any n-vertex graph $G \in \mathcal{X}_n$ the encoder assigns binary strings, called *labels*, of length $b(n)$ to the vertices of G such that the adjacency between any pair of vertices can be inferred by the decoder only from their labels. We note that the decoder depends on the class $\mathcal{X}$, but not on the graph G. The function $b(\cdot)$ is the *size* of the labeling scheme. Adjacency labeling schemes were introduced by Kannan, Naor, and Rudich [35, 36], and independently by Muller [41] in the late 1980s and have been actively studied since then.

The binary word, obtained by concatenating labels of the vertices of a graph $G \in \mathcal{X}_n$ assigned by an adjacency labeling scheme, uniquely determines graph G. Thus, a $b(n)$-bit labeling scheme cannot represent more than $2^{nb(n)}$ graphs on n vertices, and therefore, if $\mathcal{X}$ admits a $b(n)$-bit labeling scheme, then $|\mathcal{X}_n| \leqslant 2^{nb(n)}$. This implies a lower bound of $\frac{\log |\mathcal{X}_n|}{n}$ on the size $b(n)$ of any adjacency labeling scheme for $\mathcal{X}$. We say that a graph class $\mathcal{X}$ admits an *implicit representation*, if it admits an information-theoretic *order optimal* adjacency labeling scheme, i.e., if $\mathcal{X}$ has a $b(n)$-bit labeling scheme, where $b(n) = O(\log |\mathcal{X}_n|/n)$. A natural and important question is: which classes admit an adjacency labeling scheme of a size that matches this information-theoretic lower bound?

Of particular interest is the case of adjacency labeling schemes of size $O(\log n)$. This is because, under the natural assumption that vertices get assigned pairwise distinct labels, $\lceil \log n \rceil$ is a lower bound on the size of any labeling scheme. Thus, understanding the expressive power of the labeling schemes of size of order $\log n$ is a natural question.

By the above discussion adjacency labeling schemes of size $O(\log n)$ can only exist for (at most) *factorial* graph classes, i.e., classes $\mathcal{X}$ in which the number $|\mathcal{X}_n|$ of n-vertex graphs grows not faster than $2^{\Theta(n \log n)}$. It is known that the latter condition alone is not enough to guarantee adjacency labels of size $O(\log n)$ [41]. Kannan, Naor, and Rudich [35] asked if the extra restriction on $\mathcal{X}$ of being *hereditary*, i.e., closed under taking induced subgraphs, would be enough for the existence of $O(\log n)$-bit adjacency labeling scheme. This question was later stated by Spinrad [46] in the form of a conjecture, reiterated by Scheinerman [29, Chapter 6], that became known as the *Implicit Graph Conjecture*.

(*IGC*): Any hereditary factorial graph class admits an $O(\log n)$-bit labeling scheme.

The conjecture remained open for three decades until it was recently refuted in a strong form by Hatami and Hatami, who showed that for any δ there exists a hereditary factorial class that does not admit an $(n^{1/2-\delta})$-bit labeling scheme [33]. This refutation leaves wide open the question of characterizing hereditary graph classes that admit $O(\log n)$-bit labeling schemes and no plausible conjectural dichotomy is currently available.

It was recently shown that IGC does not hold even in the family of *monotone* graph classes, the hereditary classes which are closed under taking (not necessarily induced) subgraphs [11]. On the positive side, it was shown in the same work that IGC holds for every monotone *small* class, i.e., a monotone class $\mathcal{X}$ with $|\mathcal{X}_n| \leqslant n! \cdot c^n$ for some constant $c > 0$.

The hereditary small classes form a subfamily of hereditary factorial classes that contains many classes of practical or theoretical importance. For instance, forests [44], planar graphs [47], classes of bounded treewidth [8], proper minor-closes classes [43], unit interval graphs [32], classes of bounded clique-width [3], and more generally, classes of bounded

twin-width [13] are all small. All of these classes are known to admit order-optimal, and, sometimes, even asymptotically optimal adjacency labeling schemes. For example, forests admit a $(\log n + O(1))$-bit labeling scheme [5], planar graphs admit a $(1 + o(1)) \log n$-bit labeling scheme [19], classes of bounded treewidth admit a $(1+o(1)) \log n$-bit labeling scheme [28], proper minor-closed classes admit a $(2 + o(1)) \log n$-bit labeling scheme [28], graphs of clique-width at most k admit a $O(k \log k \cdot \log n)$-bit labeling scheme [46, 7], and graphs of twin-width at most k admit a $2^{2^{O(k)}} \cdot \log n$-bit labeling scheme [13]. It is known that not every hereditary (and even monotone) small class admits an asymptotically optimal labeling scheme of size $(1 + o(1)) \log n$ [10]. However, this does not rule out the existence of order optimal labeling schemes of size $O(\log n)$ for all hereditary small classes. Alon [4] recently showed that every hereditary class $\mathcal{X}$ with $|\mathcal{X}_n| = 2^{o(n^2)}$ (which include all hereditary small classes) admits an adjacency labeling scheme of size $n^{1-\varepsilon}$ for some $\varepsilon > 0$ depending on the class. To our knowledge, no better upper bound was established for hereditary small classes.

The importance of hereditary small classes and the existence of $O(\log n)$-bit labeling schemes for all monotone small classes motivated the formulation of the *Small Implicit Graph Conjecture*.

(*Small IGC*) [11]: Any hereditary small graph class admits an $O(\log n)$-bit labeling scheme.

1.1 Our contribution

In this paper we provide evidence toward the Small Implicit Graph Conjecture in two independent directions. First, we show that the Small IGC holds in the important special case of weakly sparse graph classes. Secondly, we obtain an adjacency labeling scheme of polylogarithmic size for any hereditary small class substantially improving upon the best-known polynomial-size upper bound on adjacency labeling schemes for such classes.

Our first result is a consequence of a structural property of small classes of graphs. To state this property, we need to introduce some terms. A hereditary class $\mathcal{X}$ is said to be *weakly sparse* if there exists an integer t such that the complete bipartite graph $K_{t,t}$ is not a subgraph of any graph of $\mathcal{X}$. The *degeneracy* of a graph G is the minimum number k such that every induced subgraph of G contains a vertex of degree at most k. We defer a formal definition of *bounded expansion* until Section 2 as it is somewhat technical. Instead, we note that *bounded expansion* generalizes *bounded degeneracy*, thus Theorem 1 a fortiori holds when the former is replaced by the latter.

▶ **Theorem 1.** *Every weakly sparse small class has bounded expansion.*

This structural insight into small graph classes is of independent interest due to the significance of weakly sparse graph classes. The notion of weakly sparseness is the broadest form of sparsity. It generalizes the properties of bounded degree, bounded degeneracy, and nowhere denseness [42]. By the celebrated Kővári–Sós–Turán theorem [37], among hereditary classes, weakly sparse classes are precisely the classes of graphs with a truly subquadratic number of edges. It is often observed that the extra restriction of being weakly sparse significantly simplifies the structure of graphs possessing a particular property. For example, weakly sparse graph classes of bounded clique-width have bounded treewidth [31]; weakly sparse graph classes of bounded twin-width have bounded expansion, and thus bounded degeneracy [13] (a fact that Theorem 1 generalizes, as it was proven in the same paper that every class of bounded twin-width is small); weakly sparse string graphs have bounded degeneracy [26], and even polynomial expansion [23]; weakly sparse graph classes that are well-quasi-ordered by the induced-subgraph relation have bounded pathwidth [6], etc.

A conjecture from 2016 [1], driving an ongoing program in algorithmic graph theory, suggests that, under some widely-believed complexity-theoretic assumption, a hereditary graph class admits a fixed-parameter tractable (FPT) first-order (FO) model-checking algorithm if and only if it is monadically dependent.[1] It was recently proven that every hereditary small class is monadically dependent [17] (and, in the same paper, the *only if* part of the conjecture was established). Thus, if the conjecture holds, it should in particular be true that every hereditary small class admits an FPT FO model-checking algorithm. Theorem 1 confirms that this is indeed the case within weakly sparse classes, as such an algorithm is known to exist for classes of bounded expansion [22].

Dvořák and Norine proved that if the expansion of a class is upper bounded by a sufficiently slowly-growing subexponential function, then the class is small [24]. From the definition of *expansion* given in the next section, it should be clear that the converse cannot hold since the small classes consisting of all complete graphs or of all bipartite complete graphs have unbounded expansion. A minimum requirement to establish a partial converse is thus that the class is weakly sparse. Theorem 1 confirms that under this condition the expansion of any small class is indeed bounded (albeit not necessarily by a subexponential nor a single-exponential function).

We apply Theorem 1 to obtain our first main result that the Small IGC holds for weakly sparse graph classes. This follows from a classical labeling scheme for classes of bounded degeneracy [35] (see also [11, Lemma 3.5]).

▶ **Theorem 2.** *Every weakly sparse small class admits an $O(\log n)$-bit adjacency labeling scheme.*

Our second main result holds more generally for any hereditary small class and goes some way toward the full strength of the Small IGC.

▶ **Theorem 3.** *Every hereditary small class admits an $O(\log^3 n)$-bit adjacency labeling scheme.*

This result is based on our second structural result of independent interest that hereditary small graph classes have low neighborhood complexity.

▶ **Theorem 4.** *The neighborhood complexity of any hereditary small graph class is $O(n \log n)$.*

The neighborhood complexity is a measure of structural complexity of graphs. Informally (a formal definition is given in Section 2), it reflects the diversity of the set system of the vertex neighborhoods in the graph. Low neighborhood complexity of graph classes implies nice structural and algorithmic properties. For example, in the universe of monotone graph classes, linear neighborhood complexity characterizes classes of bounded expansion [45], and almost linear neighborhood complexity characterizes nowhere dense classes [25]. Low neighborhood complexity is also used to obtain FPT algorithms for various graph problems (see for instance [25, 14]), and was recently used as a central ingredient of an FPT algorithm for the FO model-checking problem on monadically stable graph classes [16]. It is known that every hereditary graph class of bounded twin-width has linear neighborhood complexity [14] (see also [12] for improved bounds). We omit formal definitions and result statements, and refer the reader to the respective work.

[1] As with the other side remarks, we do not define the technical terms in this sentence since they will not reappear.

We obtain our Theorem 3 by combining Theorem 4 with some known result from the theory of sets systems about paths with low crossing number. In a standard form the latter result is not suitable for our need, and therefore we reformulate it and provide a self-contained proof. We believe that the result stated in this form can be of use for other needs.

1.2 Organization

The paper is organized as follows. In Section 3, we prove that weakly sparse small classes have bounded expansion (and thus bounded degeneracy); from this, we derive an $O(\log n)$-bit adjacency labeling scheme for such classes. In Section 4.1, we show that every hereditary small class has neighborhood complexity $O(n \log n)$. In Section 4.2, we use this result together with a classical result from the theory of set systems stated in a suitable form, to obtain $O(\log^3 n)$-bit labeling schemes for all hereditary small classes. In Section A, we provide a self-contained proof of the required form of the result from the theory of set systems. In the next section, we introduced necessary notation and state auxiliary facts.

2 Preliminaries

Graphs

We denote by $V(G)$ and $E(G)$ the vertex set and edge set of a graph G, respectively. When we refer to an n-vertex graph G as *labeled*, we mean that the vertex set of G is $[n]$, and we distinguish two different labeled graphs even if they are isomorphic. For a set of graphs X, we denote by $\mathrm{Lab}(X)$ the set of all labeled graphs isomorphic to a graph in X. A graph H is an *induced subgraph* of G (resp. *subgraph* of G), if it can be obtained by removing vertices of G (resp. by removing vertices and edges of G). A subgraph H of G is *spanning* if $V(H) = V(G)$. We denote by $G[S]$ the subgraph of G induced by S, i.e., obtained by removing every vertex of $V(G) \setminus S$.

The *degree* of a vertex v in G, denoted by $\deg(v)$, is the number of vertices in G adjacent to v. The *minimum* (respectively, *maximum*) *degree of* G is the minimum (respectively, maximum) degree of a vertex in G. The *average degree of* G is the ratio $\sum_{v \in V(G)} \deg(v)/|V(G)|$. Recall that the degeneracy of G is the minimum number k such that every induced subgraph of G contains a vertex of degree at most k. The following are immediate consequences of these definitions: (1) if the minimum degree of G is at least d, then the average degree of G and the degeneracy of G is at least d; (2) if the degeneracy of G is at least d, then G contains an induced subgraph of minimum degree at least d. One more relation between these three parameters is the following folklore result, which is not hard to derive by iteratively removing vertices of degree less than $d/2$.

▶ **Observation 5.** *If the average degree of a graph G is at least d, then G contains an induced subgraph of minimum degree, and thus average degree, at least $\lceil d/2 \rceil$.*

We use $G - S$ as a shorthand for $G[V(G) \setminus S]$, and $G - v$, for $G - \{v\}$. We denote by $\mathrm{Aut}(G)$ the automorphism group of a graph G, and we set $\mathrm{aut}(G) := |\mathrm{Aut}(G)|$. If X is a set of pairwise non-isomorphic n-vertex graphs, then the number of labeled graphs isomorphic to a graph in X is exactly

$$|\mathrm{Lab}(X)| = \sum_{G \in X} \frac{n!}{\mathrm{aut}(G)}. \tag{1}$$

For any non-negative integer r, the *r-subdivision* of a graph G, denoted by $r\text{-}\mathrm{subd}(G)$, is the graph obtained from G by adding one path P_{uv} on r vertices for each edge $uv \in E(G)$, and

by replacing each edge $uv \in E(G)$ by a $(r+1)$-edge path $uP_{uv}v$. In the special case were $r = 0$, r-subd(G) is the graph G. We will refer to a vertex from one of the paths P_{uv} as a *subdivision vertex* and to other vertices as *branching vertices*.

For completeness, we include a definition[2] of classes of *bounded expansion*. A class $\mathcal{C}$ has *bounded expansion* if there is a function $f : \mathbb{N} \to \mathbb{R}$ such that for every integer $r \geqslant 0$, no r-subdivision of a graph of average degree larger than $f(r)$ is a subgraph of a graph in $\mathcal{C}$. We will only apply existing results on classes of bounded expansion, and we will do so in a black-box fashion. Thus the reader should only observe that these classes are weakly sparse and of bounded average degree (by virtue of satisfying the given property for $r = 0$).

Graph Classes

A class of graphs is *hereditary* if it is closed under taking induced subgraphs, and it is *monotone* if it closed under taking subgraphs. A hereditary graph class is *small* if there exists a constant c such that the number of n-vertex *labeled* graphs in the class is at most $n! \cdot c^n$ for every $n \in \mathbb{N}$.

Graph contiguity

The *contiguity* of a graph G, denoted $\mathrm{ctg}(G)$, is the minimum integer k such that there exists a linear order of its vertices in which the neighborhood of each vertex of G can be partitioned into at most k disjoint intervals. For a function $k : \mathbb{N} \to \mathbb{N}$ and a class of graphs $\mathcal{C}$, we say that $\mathcal{C}$ has contiguity at most k, if for every n-vertex graph $G \in \mathcal{C}$ the contiguity of G is at most $k(n)$.

The idea of contiguity was introduced in the context of range searching under the name of *crossing number of spanning paths* [48] (see also Section A). Independently, it was introduced in the context of DNA reconstruction under the name of *k-consecutive ones property* [30], which generalizes the notion of *consecutive ones property* for matrices [15, 27]. More recently, the concept was utilized as part of an FPT algorithm for the FO model-checking problem on monadically stable graph classes [16], and has appeared in the context of implicit graph representations [4, 2, 9]. In particular, Alon used it to show that every hereditary graph class $\mathcal{X}$ with $|\mathcal{X}_n| = 2^{o(n^2)}$ admits a $n^{1-\varepsilon}$-bit labeling scheme for some $\varepsilon = \varepsilon(\mathcal{X}) > 0$ [4]. This was done via the following simple connection between contiguity and adjacency labeling schemes.

▶ **Proposition 6.** *Let $k : \mathbb{N} \to \mathbb{N}$ be a function, and $\mathcal{C}$ be a class of graphs of contiguity at most k. Then $\mathcal{C}$ admits an adjacency labeling scheme of size at most $(2k(n) + 1) \log n$.*

Proof. Let G be a n-vertex graph in $\mathcal{C}$. To construct adjacency labels for G of the target size, we fix a linear ordering σ of the vertices of G witnessing contiguity at most $k(n)$. The label of a vertex v of G then consists of its position in σ, followed by the positions of the endpoints of the at most $k(n)$ intervals describing the neighborhood of v. The decoder infers adjacency between two vertices u and v by testing if the position of u in σ belongs to one of the intervals describing the neighborhood of v. ◀

Set systems, shatter functions, and neighborhood complexity

A pair $(X, \mathcal{S})$, where X is a finite set and $\mathcal{S}$ is a family of subsets of X is called a set system. The *incidence matrix* M of the set system $(X, \mathcal{S})$ is the matrix with $|X|$ columns and $|\mathcal{S}|$ rows indexed by the elements of X and $\mathcal{S}$, respectively, where M_{ij} is 1 if the i-th set of $\mathcal{S}$ contains the j-th element of X, and 0 otherwise.

[2] Actually, we rather give an established characterization [20] as it is more compact.

The *primal shatter function* of a set system $(X, \mathcal{S})$ is the function $\pi_{\mathcal{S}}$ given by

$$\pi_{\mathcal{S}}(m) = \max_{A \subseteq X,\, |A|=m} |\{Y \cap A \ : \ Y \in \mathcal{S}\}|.$$

The set system $(X^*, \mathcal{S}^*)$, where $X^* = \mathcal{S}$, $\mathcal{S}^* = \{S_x \ : \ x \in X\}$, and $S_x = \{S \in \mathcal{S} \ : \ x \in X\}$, is called the dual set system of $(X, \mathcal{S})$. The shatter function of $(X^*, \mathcal{S}^*)$ is called the *dual shatter function* of $(X, \mathcal{S})$. Note that if M is the incidence matrix of (X, S), then the transpose of M is the incidence matrix of $(X^*, \mathcal{S}^*)$. A subset $A \subseteq X$ is *shattered* by $\mathcal{S}$ if every subset B of A can be obtained as the intersection $B = A \cap Y$ for some $Y \in \mathcal{S}$, i.e., $\{Y \cap A \mid Y \in \mathcal{S}\} = 2^A$. The *Vapnik–Chervonenkis dimension* (or *VC dimension* for short) of $\mathcal{S}$ is the maximum of the sizes of all shattered subsets of X.

Given a vertex v of a graph G, we use standard notation $N_G(v)$ to denote the set $\{u : vu \in E(G)\}$ of neighbors of v in G. The *neighborhood set system* of G is the set system $(V(G), \mathcal{N}_G)$ where

$$\mathcal{N}_G = \{N_G(v) : v \in V(G)\}.$$

Observe that the adjacency matrix of G is the incidence matrix of $(V(G), \mathcal{N}_G)$. Thus, since the adjacency matrix of an undirected graph is symmetric, the primal and the dual shatter functions of a neighborhood set system coincide.

▶ **Observation 7.** *For neighborhood set systems, the primal and dual shatter functions coincide. That is, for any n-vertex graph G,*

$$\pi_{\mathcal{N}_G}(m) = \pi^*_{\mathcal{N}_G}(m) \qquad \text{for all } m \in [n].$$

We define the *neighborhood complexity* of a graph G, denoted by ν_G, as the primal (equivalently, dual) shatter function of its neighborhood set system, i.e., $\nu_G(m) = \pi_{\mathcal{N}_G}(m) = \pi^*_{\mathcal{N}_G}(m)$ for all $m \in [|V(G)|]$.

Given a graph class $\mathcal{C}$, the *neighborhood complexity* of $\mathcal{C}$ is the function $\nu_{\mathcal{C}} : \mathbb{N} \to \mathbb{N}$ defined by

$$\nu_{\mathcal{C}}(n) := \sup_{G \in \mathcal{C},\, A \subseteq V(G),\, |A|=n} |\{N(v) \cap A \ : \ v \in V(G)\}| = \sup_{G \in \mathcal{C}} \pi_{\mathcal{N}(G)}(n) = \sup_{G \in \mathcal{C}} \nu_G(n).$$

3 Implicit representation for weakly sparse small classes

In this section we prove Theorem 1, which states that any weakly sparse small class has bounded expansion, and hence, in particular, bounded degeneracy. Together with the known labeling scheme for bounded degeneracy graphs (see e.g. [11, Lemma 3.5]), this immediately implies that any weakly sparse small class admits an $O(\log n)$-bit adjacency labeling scheme (Theorem 2).

The proof of Theorem 1 relies on the fact (Corollary 10) that for any weakly sparse class $\mathcal{X}$ of unbounded expansion there exists an integer $r \geqslant 1$ such that $\mathcal{X}$ contains r-subdivisions of graphs of arbitrarily large average degree. This is a consequence of the forward direction of a characterization of classes of (un)bounded expansion by Dvořák [21] and a classical result due to Kühn and Osthus [39], both stated below.

▶ **Theorem 8** ([21, Theorem 5, $(a) \Leftrightarrow (c)$])**.** *A hereditary graph class $\mathcal{X}$ has unbounded expansion if and only if there is an integer $r \geqslant 0$ such that for any integer d, $\mathcal{X}$ contains the r-subdivision of a graph of average degree at least d.*

▶ **Theorem 9** ([39, Theorem 2]). *For all $d, t \in \mathbb{N}$, there exists an integer $D := D(d,t)$ such that every $K_{t,t}$-free graph G of average degree at least D contains $1\text{-subd}(F)$ as an induced subgraph for some graph F of average degree at least d.*

We now extract the announced fact, of which we will only need the *only if* direction.

▶ **Corollary 10.** *A weakly sparse graph class $\mathcal{X}$ is of unbounded expansion if and only if there is an integer $r \geqslant 1$ such that for any integer d, $\mathcal{X}$ contains the r-subdivision of a graph of average degree at least d.*

Proof. The *if* part follows directly from Theorem 8. For the *only if* part, we first apply the corresponding implication in Theorem 8. If $r \geqslant 1$, we are done. If instead $r = 0$, then $\mathcal{X}$ contains graphs of arbitrarily large average degree. Therefore, Theorem 9 implies that $\mathcal{X}$ contains 1-subdivisions of graphs of arbitrarily large average degree, i.e., the sought statement holds for $r = 1$. ◀

Equipped with this tool, the proof of Theorem 1 shows the contrapositive as follows. Assuming a weakly sparse class $\mathcal{X}$ has unbounded expansion, we fix a positive integer r such that $\mathcal{X}$ contains r-subdivisions of graphs of arbitrarily large average degree, which exists by Corollary 10. Let F be a graph with large average degree such that the r-subdivision of F is in $\mathcal{X}$. First, we extract from F a subgraph H with large minimum degree that contains a bounded-degree spanning tree. This is possible due to Observation 5 and the following lemma.

▶ **Lemma 11** ([11, Lemma 3.3]). *Let F be a graph of minimum degree d. Then F has an induced subgraph H of minimum degree at least d with a spanning tree of* maximum degree *at most d.*

Next, we use the properties of H to show that it contains many (not necessarily induced) subgraphs each with a small number of automorphisms, which are translated to many *induced* subgraphs with a small number of automorphisms in the r-subdivision of H and thus in the r-subdivision of the original graph F. This allows us to lower bound the number of labeled graphs on the same number of vertices in $\mathcal{X}$ by counting subgraphs of H, and finally conclude that $\mathcal{X}$ is not small. The quantitative connection is made possible by the following observations.

▶ **Observation 12.** *For any integer $r \geqslant 0$, two graphs G and H are isomorphic if and only if $r\text{-subd}(G)$ and $r\text{-subd}(H)$ are isomorphic.*

▶ **Observation 13.** *Let F be a graph. For any integer $r > 0$, for every subgraph H of F, the graph $r\text{-subd}(H)$ is isomorphic to an induced subgraph of $r\text{-subd}(F)$.*

▶ **Observation 14.** *Let G be a connected graph containing a vertex of degree at least 3, then*

$$\mathrm{aut}(r\text{-subd}(G)) = \mathrm{aut}(G).$$

Proof. We start by showing that any automorphism φ of $r\text{-subd}(G)$ maps branching vertices to branching vertices and subdivision vertices to subdivision vertices. First, note that as G has a vertex of degree at least 3, the graph $r\text{-subd}(G)$ also has a vertex of degree at least 3 (hence a branching vertex) say u, which must be mapped by φ to a vertex of degree at least 3 (hence, to a branching vertex), say v. Now, since G is connected, the branching vertices are exactly the ones that are at distances 0 (mod $r+1$) from u, and they are mapped to

vertices that are at distances 0 (mod $r+1$) from v, and thus, to branching vertices. Hence, the restriction $\tilde{\varphi}$ of φ to the branching vertices of r-subd(G) is a permutation of the vertices of G. We claim that $\tilde{\varphi}$ is an automorphism of G. Indeed, two vertices a and b are adjacent in G if and only if a and b are at distance $r+1$ in r-subd(G) if and only if $\varphi(a)$ and $\varphi(b)$ are at distance $r+1$ in r-subd(G) if and only if $\tilde{\varphi}(a) = \varphi(a)$ and $\tilde{\varphi}(b) = \varphi(b)$ are adjacent in G.

From this, we deduce that $\varphi \mapsto \tilde{\varphi}$ is a bijection between the automorphisms of r-subd(G) and the automorphisms of G: it is injective since an automorphism of r-subd(G) is uniquely determined by the images of the branching vertices, and surjective since there are at least as many automorphisms of r-subd(G) as there are of G. Hence, $\mathrm{aut}(r\text{-subd}(G)) = \mathrm{aut}(G)$. ◀

Before proceeding to our main result, we provide some basic facts about automorphisms, which we will rely on. Given two graphs F and G, we denote by $\#\mathrm{Sub}(F \to G)$ the number of subgraphs of G isomorphic to F, and by $\#\mathrm{Emb}(F \to G)$ the number of embeddings of F into G, i.e., the number of injective homomorphisms from F to G. Thus

$$\#\mathrm{Emb}(F \to G) = \#\mathrm{Sub}(F \to G) \cdot \mathrm{aut}(F).$$

For instance, if G is the cycle on an even number n of vertices, and F is the disjoint union of $n/2$ edges, then $\#\mathrm{Sub}(F \to G) = 2$ and $\mathrm{aut}(F) = (n/2)! \cdot 2^{\frac{n}{2}}$, so $\#\mathrm{Emb}(F \to G) = (n/2)! \cdot 2^{\frac{n}{2}+1}$.

We will use the following two well known facts (see [38]).

▶ **Lemma 15.** *Let F be a spanning subgraph of a graph G. Then*

$$\mathrm{aut}(G) \leqslant \#\mathrm{Emb}(F \to G) = \#\mathrm{Sub}(F \to G) \cdot \mathrm{aut}(F).$$

▶ **Lemma 16.** *Let G be a connected graph of maximum degree Δ. Then*

$$\mathrm{aut}(G) \leqslant n \cdot \Delta! \cdot (\Delta - 1)^{n-\Delta-1} \leqslant n\Delta^n.$$

We apply the above two facts to upper bound the number of automorphisms of a graph containing a bounded-degree spanning tree in terms its average degree.

▶ **Lemma 17.** *Let $d \geqslant 1$ and G be a connected n-vertex graph with at most dn edges, which has a spanning tree T of maximum degree τ. Then* $\mathrm{aut}(G) \leqslant (6\tau d)^n$.

Proof. By Lemma 16, $\mathrm{aut}(T) \leqslant n\tau^n \leqslant (2\tau)^n$. Moreover, $\#\mathrm{Sub}(T \to G)$ is at most the number of ways of choosing $n-1$ edges among $E(G)$. Hence

$$\#\mathrm{Sub}(T \to G) \leqslant \binom{dn}{n-1} \leqslant \left(\frac{edn}{n-1}\right)^{n-1} = \left(1 + \frac{1}{n-1}\right)^{n-1} \cdot (ed)^{n-1} \leqslant e(ed)^{n-1} \leqslant (ed)^n,$$

where in the penultimate inequality we used the fact that $1 + x \leqslant e^x$ for any real x. Hence, by Lemma 15, we have $\mathrm{aut}(G) \leqslant \#\mathrm{Sub}(T \to G) \cdot \mathrm{aut}(T) \leqslant (ed)^n \cdot (2\tau)^n \leqslant (6\tau d)^n$. ◀

We are now ready to prove the main result of this section.

▶ **Theorem 1.** *Every weakly sparse small class has bounded expansion.*

Proof. For the sake of contradiction, suppose that a weakly sparse graph class $\mathcal{X}$ has unbounded expansion, but there exists a constant c such that for every $n \in \mathbb{N}$ the number of labeled n-vertex graphs in $\mathcal{X}$ is at most $n! \cdot c^n$. Let $r \geqslant 1$ be an integer such that $\mathcal{X}$ contains r-subdivisions of graphs of arbitrary large average degree, which exists by Corollary 10.

Let $\delta := 24$, $d := \lceil \max(\delta^2, \delta c^{2r}) \rceil$. Let F be a graph of average degree at least $4d$, such that r-subd(F) is in $\mathcal{X}$, and let F' be a subgraph of F of minimum degree at least $2d$, which exists by Observation 5. By Lemma 11, F' contains a subgraph H' of minimum degree at

least $2d$ with a spanning tree T of maximum degree $2d$. If H' has more than $d|V(H')|$ edges, we remove all but $d|V(H')| - (|V(H')| - 1)$ of them from the set $E(H') \setminus E(T)$ to obtain a spanning subgraph with exactly $d|V(H')|$ edges that still contains T as a spanning tree. We denote this subgraph by H and let $k := |V(H)|$, and thus $|E(H)| = dk$.

Let $\mathcal{H}$ be the set of spanning subgraphs of H that contain T as a subgraph and have exactly δk edges. Then, since $\delta \leqslant d$, we have

$$|\mathcal{H}| = \binom{dk-(k-1)}{\delta k-(k-1)} \geqslant \binom{dk-k}{\delta k-k} \geqslant \left(\frac{d-1}{\delta-1}\right)^{(\delta-1)k} \geqslant \left(\frac{d}{\delta}\right)^{(\delta-1)k}.$$

Since every graph in $\mathcal{H}$ has $\delta k \leqslant dk$ edges and contains a spanning tree of maximum degree $2d$, from Lemma 17,

$$\max_{A\in\mathcal{H}} \operatorname{aut}(A) \leqslant (6 \cdot 2d \cdot d)^k \leqslant (4d)^{2k}. \tag{2}$$

Thus, denoting by $\mathcal{U}$ the set of pairwise non-isomorphic graphs in $\mathcal{H}$, we obtain

$$|\mathcal{U}| \geqslant \frac{|\mathcal{H}|}{\max_{A\in\mathcal{H}} \operatorname{aut}(A)} \geqslant \left(\frac{d}{\delta}\right)^{(\delta-1)k} \cdot \frac{1}{(4d)^{2k}} = \left(\frac{d}{\delta}\right)^{(\delta+1)k} \cdot \frac{\delta^{2k}}{4^{2k} \cdot d^{4k}} = \left(\frac{d}{\delta}\right)^{(\delta+1)k} \cdot \frac{6^{2k}}{d^{4k}} \tag{3}$$

Let $r\text{-}\operatorname{subd}(\mathcal{U})$ be the set of r-subdivisions of graphs in $\mathcal{U}$, and observe that every graph in $r\text{-}\operatorname{subd}(\mathcal{U})$ has exactly $N := (r\delta+1)k$ vertices. Since every graph in $\mathcal{U}$ is a subgraph of F, and $r\text{-}\operatorname{subd}(F)$ is in $\mathcal{X}$, Observation 13 implies that every graph in $r\text{-}\operatorname{subd}(\mathcal{U})$ is in $\mathcal{X}$. Thus, to contradict the assumption that, for every $n \in \mathbb{N}$, class $\mathcal{X}$ has at most $n! \cdot c^n$ labeled n-vertex graphs, it is enough to show that $|\operatorname{Lab}(r\text{-}\operatorname{subd}(\mathcal{U}))| > c^N \cdot N!$. This is what we do in the remainder of the proof.

It follows from Observation 12 that $r\text{-}\operatorname{subd}(\mathcal{U})$ is a set of pairwise non-isomorphic graphs and $|r\text{-}\operatorname{subd}(\mathcal{U})| = |\mathcal{U}|$. Furthermore, since every graph A in $\mathcal{H}$ is connected and has k vertices and $\delta k \geqslant k+1$ edges, it contains a vertex of degree at least 3 and thus, by Observation 14, $\operatorname{aut}(r\text{-}\operatorname{subd}(A)) = \operatorname{aut}(A)$. Therefore, from Equation (1), we have

$$|\operatorname{Lab}(r\text{-}\operatorname{subd}(\mathcal{U}))| \geqslant |r\text{-}\operatorname{subd}(\mathcal{U})| \cdot \frac{N!}{\max_{A\in\mathcal{H}} \operatorname{aut}(r\text{-}\operatorname{subd}(A))} = |\mathcal{U}| \cdot \frac{N!}{\max_{A\in\mathcal{H}} \operatorname{aut}(A)},$$

and thus, from (2) and (3), and the fact that $N/r = (\delta + 1/r)k \leqslant (\delta+1)k$, we obtain

$$|\operatorname{Lab}(r\text{-}\operatorname{subd}(\mathcal{U}))| \geqslant \left(\frac{d}{\delta}\right)^{N/r} \cdot \frac{6^{2k}}{d^{4k}} \cdot \frac{N!}{(4d)^{2k}} > \left(\frac{d}{\delta}\right)^{N/r} \cdot \frac{N!}{d^{6k}},$$

Finally, since $\frac{N}{4r} = \frac{(r\delta+1)k}{4r} > \frac{\delta k}{4} = 6k$, and d satisfies both $\sqrt{d} \geqslant \delta$ and $(d/\delta)^{1/r} \geqslant c^2$, we have

$$|\operatorname{Lab}(r\text{-}\operatorname{subd}(\mathcal{U}))| > \left(\frac{d}{\delta}\right)^{\frac{N}{2r}} \cdot \left(\frac{d}{\delta}\right)^{\frac{N}{2r}} \cdot \frac{N!}{d^{6k}} \geqslant \left(\frac{d}{\delta}\right)^{\frac{N}{2r}} \cdot d^{\frac{N}{4r}} \cdot \frac{N!}{d^{6k}} > \left(\left(\frac{d}{\delta}\right)^{1/r}\right)^{\frac{N}{2}} \cdot N! \geqslant c^N \cdot N!,$$

which completes the proof. ◀

4 Short adjacency labels for small classes

In this section we prove:

▶ **Theorem 3.** *Every hereditary small class admits an $O(\log^3 n)$-bit adjacency labeling scheme.*

To do so, we first show a result of independent interest (in Section 4.1), that the neighborhood complexity of any small class is at most $O(n \log n)$. Then, we prove in Section 4.2 that such neighborhood complexity implies $O(\log^2 n)$ contiguity. The latter together with Proposition 6 implies Theorem 3. The core part of this proof strategy is to show how the bound on the neighborhood complexity can be translated to the bound on contiguity. This is done via a classical result from the theory of set systems of bounded Vapnik-Chervonenkis dimension on so-called *paths with low crossing number* (Theorem 18) applied to neighborhood set systems. While the result about paths with low crossing number is a known fact, we need it in somewhat unusual form that we could not find elsewhere in the literature. Therefore, in Section A we present a self-contained proof of this result in a suitable form.

4.1 Neighborhood complexity of small classes

In this section we show that small classes have low neighborhood complexity.

▶ **Theorem 4.** *The neighborhood complexity of any hereditary small graph class is $O(n \log n)$.*

Proof. Let $\mathcal{C}$ be a hereditary small class of graphs and $c \geqslant 1$ be a constant such that $|\mathcal{C}_n| \leqslant n! \cdot c^n$ holds for every $n \in \mathbb{N}$. To prove the claim, we will show that for every graph H in $\mathcal{C}$ it holds that

$$\nu_H(m) < 9c^2 m \log(m+1) \qquad \text{for all } m \in [|V(H)|].$$

Suppose this is not the case. Then, by definition, there exists a graph $H \in \mathcal{C}$ and an n-element set $A \subseteq V(H)$ such that $|\{N_H(v) \cap A \ : \ v \in V(H)\}| \geqslant 9c^2 n \log(n+1)$. Note that the latter inequality implies $n \geqslant 4$. Let $B' \subseteq V(H)$ be any maximum-size set of vertices whose neighborhoods have pairwise distinct intersections with A in the graph H, and let $B := B' \setminus A$. Denote by G the subgraph of H induced by $A \cup B$. By the assumption, we have

1. $|A| = n$,
2. $|B| \geqslant |B'| - |A| \geqslant 9c^2 n \log(n+1) - n > 8c^2 n \log n$,
3. $A \cap B = \emptyset$, and
4. all vertices in B have pairwise distinct neighborhoods in A in graph G.

Let $n_1 := n\lfloor \log n \rfloor$ and $N := n_1 + n$. We will show that the hereditary closure of G contains more than $N! c^N$ labeled graphs on N vertices, i.e., that $|\mathrm{Lab}(\mathrm{IndSub}_N(G))| > N! c^N$, where $\mathrm{IndSub}_N(G)$ is the set of N-vertex induced subgraphs of G. This would contradict our assumption on the number of graphs in $\mathcal{C}$. To do this, let us first fix a labeling of A using the integers from 1 to n. We claim that any two distinct labeled subsets $X_1, X_2 \subseteq B$ of size n_1, labeled with the integers from $n+1$ to N, induce two distinct labeled graphs $G_1 = G[A \cup X_1]$ and $G_2 = G[A \cup X_2]$ contained in $\mathcal{C}$. Indeed, since all vertices in B have pairwise distinct neighborhoods in A, G_1 and G_2 contain a vertex with the same label in $[n+1, N]$ but different neighborhoods in A (i.e. different neighborhoods among vertices with labels from $[n]$).

Hence, the number of N-vertex labeled graphs in $\mathcal{C}$ that are induced subgraphs of G is lower bounded by the number of ways to choose a labeled subset of size n_1 from B, which is

$$n_1! \cdot \binom{|B|}{n_1} \geqslant n_1! \cdot \left(\frac{|B|}{n_1}\right)^{n_1} > n_1! \cdot (8c^2)^{n_1}. \tag{4}$$

We want to show that this is more than the total number of N-vertex labeled graphs in $\mathcal{C}$, which, by assumption, does not exceed

$$N! \cdot c^N = (n_1+n)! \cdot c^{n_1+n} \leqslant n_1! \cdot (n_1+n)^n \cdot c^{2n_1}. \tag{5}$$

From Equation (4) and Equation (5), we should then show that

$$n_1! \cdot (8c^2)^{n_1} \geqslant n_1! \cdot (n_1 + n)^n \cdot c^{2n_1},$$

which is equivalent to establishing that

$$8^{n_1} = 2^{3n_1} \geqslant (n_1 + n)^n = 2^{n \log(n_1+n)}.$$

The latter indeed holds since

$$\begin{aligned} n \log(n_1 + n) &\leqslant n \log(2n\lfloor \log n \rfloor) \\ &\leqslant n \log 2 + n \log n + n \log\lfloor \log n \rfloor \\ &\leqslant n(1 + \log 2) + n\lfloor \log n \rfloor + n \log\lfloor \log n \rfloor \\ &\leqslant n\lfloor \log n \rfloor + n\lfloor \log n \rfloor + n \log\lfloor \log n \rfloor \\ &\leqslant n_1 + n_1 + n_1 = 3n_1, \end{aligned}$$

where the penultimate inequality holds because $n \geqslant 4$. ◀

4.2 From neighborhood complexity to contiguity

In this section, we first show how to translate a bound on neighborhood complexity to a bound on contiguity (Theorem 20), and then apply it to obtain bounds on contiguity for hereditary small classes (Theorem 21). A crucial technical tool that we use to establish this translation is a suitable form of a known result about paths with low crossing number (Theorem 18). To state this result we must first introduce the required notions.

Let $(X, \mathcal{S})$ be a set system. A set $S \in \mathcal{S}$ *crosses* a 2-element set $\{x, y\} \subset X$ if exactly one of x and y belongs to S. Let $\mathcal{F}$ be a multiset of 2-element subsets in X. The *crossing number of $\mathcal{F}$ with respect to a set $S \in \mathcal{S}$* is the number of elements in $\mathcal{F}$ that are crossed by S. The *crossing number of $\mathcal{F}$ with respect to $\mathcal{S}$* is the maximum crossing number of $\mathcal{F}$ with respect to a set $S \in \mathcal{S}$. If $(X, \mathcal{F})$, considered as a graph, is a path spanning all elements of X, then we say that $(X, \mathcal{F})$ is a path on X. In this case, the crossing number of $\mathcal{F}$ is referred to as the crossing number of the corresponding path. We are now ready to state the main technical tool whose self-contained proof can be found in Section A.

▶ **Theorem 18.** *Let X be an n-element set, $f : \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{\geqslant 0}$ be a strictly increasing function, and d be a natural number. Let $(X, \mathcal{S})$ be a set system of VC dimension d that satisfies $\pi^*_{\mathcal{S}}(m) \leqslant f(m)$ for all $m \in [n]$. Then there exists a path on X with crossing number at most*

$$2 \log |\mathcal{S}| + 10d \cdot \sum_{j=1}^{n} \frac{1}{f^{-1}(j/2)}.$$

We will apply Theorem 18 to neighborhood set systems. Its importance in our context is due to (a) the fact that the dual shatter function of the neighborhood set system of a graph coincides with its primal dual function (Observation 7), i.e., the neighborhood complexity of the graph; and (b) the following observation about close relationship between the crossing number of a path for the neighborhood set system and the contiguity of the graph.

▶ **Observation 19.** *Let G be an n-vertex graph. If there exists a path on $V(G)$ that has crossing number with respect to $\mathcal{N}_G$ at most k, then the contiguity of G is at most $k/2 + 1$.*

Proof. Let $\{\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}\}$ be a path on $V(G)$ with crossing number with respect to $\mathcal{N}_G$ at most k. We claim that the linear ordering $\sigma = v_1, v_2, \ldots, v_n$ of $V(G)$ witnesses the target bound on the contiguity of G.

Let v be an arbitrary vertex in G and let N be its neighborhood. For convenience, mark a vertex of G by 1 if it belongs to N and 0 otherwise. This marking together with the ordering σ correspond to the binary word $w = w_1 w_2 \ldots w_n$, where w_i is the mark of v_i for every $i \in [n]$. A relation between pairs crossed and neighborhoods partitioned now follows from two observations:

1. in the word w, the maximal intervals of consecutive 1s correspond to intervals of consecutive vertices in σ that partition the neighborhood N of v;
2. each pair of consecutive letters $w_i w_{i+1}$ in w with $w_i \neq w_{i+1}$ corresponds to a crossing of $\{v_i, v_{i+1}\}$ by N.

To conclude, each interval in the partition of N corresponds to two crossings apart from the at most two intervals at each end of the ordering which each corresponds to one crossing. Thus the number of intervals is at most $2 + \lfloor \frac{k-2}{2} \rfloor \leqslant k/2 + 1$, as claimed. ◀

The next theorem translates a bound on neighborhood complexity to a bound on contiguity. It follows directly from Theorem 18, Observation 19, Observation 7, and the definition of neighborhood complexity.

▶ **Theorem 20.** *Let G be an n-vertex graph, $f : \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{\geqslant 0}$ be a strictly increasing function, and d be a natural number. If $(V, \mathcal{N}_G)$ has VC dimension d and $\nu_G(m) \leqslant f(m)$ for all $m \in [n]$, then*

$$\mathrm{ctg}(G) \leqslant 1 + \log n + 5d \cdot \sum_{j=1}^{n} \frac{1}{f^{-1}(j/2)}.$$

We now apply Theorem 20 to hereditary small classes, which have $O(n \log n)$ neighborhood complexity by Theorem 4, to obtain a bound on their contiguity in closed form.

▶ **Theorem 21.** *Let $\mathcal{X}$ be a hereditary small class, then the contiguity of $\mathcal{X}$ is $O(\log^2 n)$.*

Proof. Since $\mathcal{X}$ is a small class, by Theorem 4 there exists some constant C such that $\nu_{\mathcal{X}} \leqslant f(n)$ where $f(n) := Cn \log n$. Consequently, for every $n \in \mathbb{N}$ and every n-vertex graph $G \in \mathcal{X}$, we have $\pi_{\mathcal{N}_G}(m) \leqslant f(m)$ for all $m \in [n]$. Thus, by definition, the VC dimension d of $(V(G), \mathcal{N}_G)$ satisfies $2^d \leqslant Cd \log d$, which implies

$$d \leqslant \log C + \log d + \log\log d \leqslant \log C + 4d/5,$$

and therefore $d \leqslant 4 \log C$.

Now, by Theorem 20, we have

$$\mathrm{ctg}(G) \leqslant 1 + \log n + 20 \cdot \log C \cdot \sum_{j=1}^{n} \frac{1}{f^{-1}(j)}. \tag{6}$$

Since f is increasing on $[1, \infty)$ with image $[0, \infty)$, its inverse f^{-1} exists on $[0, \infty)$, and is also increasing. The exact inverse f^{-1} is somewhat complicated, in particular $f^{-1}(1) = e^{W(\ln(2)/C)}$, where $W(\cdot)$ is the Lambert function. We consider $f^{-1}(1) > 0$ just as a constant depending on C, and show that

$$f^{-1}(x) \geqslant \frac{x}{C \log x} \qquad \text{for all } x \geqslant 2. \tag{7}$$

To show (7), since $\log x \geqslant 1$ for all $x \geqslant 2$ and we can assume $C \geqslant 1$, we have

$$f\left(\frac{x}{C\log x}\right) = C \cdot \frac{x}{C\log x} \cdot \log\left(\frac{x}{C\log x}\right) \leqslant x. \tag{8}$$

Thus, applying f^{-1} to both sides of (8) gives $f^{-1}(x) \geqslant \frac{x}{C\log x}$, establishing (7).

Then, by (6),

$$\begin{aligned}\mathrm{ctg}(G) &\leqslant 1 + \log n + 20 \cdot \log C \cdot \sum_{j=1}^{n} \frac{1}{f^{-1}(j)} \\ &\leqslant 1 + \log n + 20 \cdot \log C \cdot e^{-W(\ln(2)/C)} + 20 \cdot \log C \cdot \sum_{j=2}^{n} \frac{C\log j}{j} \\ &= \Theta(\log^2 n),\end{aligned}$$

as claimed. ◀

The main Theorem 3 of this section is a consequence of Theorems 4 and 21 and Proposition 6.

5 Conclusion

In this paper we obtained strong evidence in support of the Small Implicit Graph Conjecture that posits that every hereditary small class admits an $O(\log n)$-bit adjacency labeling scheme. Specifically, we showed that (1) every weakly sparse small class admits such a labeling scheme; (2) every hereditary small class admits an $O(\log^3 n)$-bit adjacency labeling scheme. To obtain these results we established two structural properties of hereditary small classes that are of independent interest. The first property is that every weakly sparse small class has bounded degeneracy, and even, bounded expansion. The second property is that every hereditary small class has neighborhood complexity $O(n \log n)$.

The latter property leaves a tantalizing open question of whether the bound on the neighborhood complexity can be further improved. All hereditary small classes known to us have linear neighborhood complexity (e.g., classes of bounded twin-width [14, 12]), which motivates the following:

▶ **Conjecture 22.** *Every hereditary small class of graphs has neighborhood complexity $O(n)$.*

If this conjecture is true, our approach would imply $O(\log^2 n)$-bit labeling schemes for all hereditary small classes.

References

1 Algorithms, Logic and Structure Workshop in Warwick – Open Problem Session, 2016. URL: https://warwick.ac.uk/fac/sci/maths/people/staff/daniel_kral/alglogstr/openproblems.pdf.

2 Bogdan Alecu, Vladimir E. Alekseev, Aistis Atminas, Vadim Lozin, and Viktor Zamaraev. Graph parameters, implicit representations and factorial properties. *Discrete Mathematics*, 346(10):113573, 2023. doi:10.1016/j.disc.2023.113573.

3 Peter Allen, Vadim V. Lozin, and Michaël Rao. Clique-width and the speed of hereditary properties. *Electron. J. Comb.*, 16(1), 2009. doi:10.37236/124.

4 Noga Alon. Implicit representation of sparse hereditary families. *Discret. Comput. Geom.*, to appear, 2023. doi:10.1007/s00454-023-00521-0.

5 Stephen Alstrup, Søren Dahlgaard, and Mathias Bæk Tejs Knudsen. Optimal induced universal graphs and adjacency labeling for trees. *Journal of the ACM (JACM)*, 64(4):1–22, 2017. doi:10.1145/3088513.

6 Aistis Atminas, Vadim V Lozin, and Igor Razgon. Graphs without large bicliques and well-quasi-orderability by the induced subgraph relation. *Journal of Combinatorics*, 10(2):327–337, 2019. doi:10.4310/JOC.2019.v10.n2.a8.

7 Avah Banerjee. An adjacency labeling scheme based on a decomposition of trees into caterpillars. In *Combinatorial Algorithms: 33rd International Workshop, (IWOCA 2022), proceedings*, pages 114–127. Springer, 2022. doi:10.1007/978-3-031-06678-8_9.

8 Lowell W Beineke and Raymond E Pippert. The number of labeled k-dimensional trees. *Journal of Combinatorial Theory*, 6(2):200–205, 1969.

9 Édouard Bonnet, Julien Duron, John Sylvester, and Viktor Zamaraev. Symmetric-Difference (Degeneracy) and Signed Tree Models. In *49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024)*, volume 306 of *LIPIcs*, pages 32:1–32:16, 2024. doi:10.4230/LIPIcs.MFCS.2024.32.

10 Édouard Bonnet, Julien Duron, John Sylvester, Viktor Zamaraev, and Maksim Zhukovskii. Small but unwieldy: A lower bound on adjacency labels for small classes. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2024)*, pages 1147–1165, 2024. doi:10.1137/1.9781611977912.44.

11 Édouard Bonnet, Julien Duron, John Sylvester, Viktor Zamaraev, and Maksim Zhukovskii. Tight bounds on adjacency labels for monotone graph classes. In *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024*, volume 297 of *LIPIcs*, pages 31:1–20, 2024. doi:10.4230/LIPIcs.ICALP.2024.31.

12 Édouard Bonnet, Florent Foucaud, Tuomo Lehtilä, and Aline Parreau. Neighbourhood complexity of graphs of bounded twin-width. *European Journal of Combinatorics*, 115:103772, 2024. doi:10.1016/j.ejc.2023.103772.

13 Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width II: small classes. *Combinatorial Theory, 2 (2)*, 2022. doi:10.5070/c62257876.

14 Édouard Bonnet, Eun Jung Kim, Amadeus Reinald, Stéphan Thomassé, and Rémi Watrigant. Twin-width and polynomial kernels. *Algorithmica*, 84(11):3300–3337, 2022. doi:10.1007/s00453-022-00965-5.

15 Kellogg S Booth and George S Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms. *Journal of computer and system sciences*, 13(3):335–379, 1976. doi:10.1016/S0022-0000(76)80045-1.

16 Jan Dreier, Ioannis Eleftheriadis, Nikolas Mählmann, Rose McCarty, Michał Pilipczuk, and Szymon Toruńczyk. First-order model checking on monadically stable graph classes. *arXiv preprint*, 2023. arXiv:2311.18740.

17 Jan Dreier, Nikolas Mählmann, and Szymon Toruńczyk. Flip-breakability: A combinatorial dichotomy for monadically dependent graph classes. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, pages 1550–1560, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3618260.3649739.

18 R. M. Dudley. Central limit theorems for empirical measures. *Ann. Probab.*, 6(6):899–929, 1978.

19 Vida Dujmović, Louis Esperet, Cyril Gavoille, Gwenaël Joret, Piotr Micek, and Pat Morin. Adjacency labelling for planar graphs (and beyond). *Journal of the ACM (JACM)*, 68(6):1–33, 2021. doi:10.1145/3477542.

20 Zdeněk Dvořák. Asymptotical structure of combinatorial objects. *PhD thesis*, 2007.

21 Zdeněk Dvořák. Induced subdivisions and bounded expansion. *Eur. J. Comb.*, 69:143–148, 2018. doi:10.1016/J.EJC.2017.10.004.

22 Zdenek Dvorák, Daniel Král, and Robin Thomas. Testing first-order properties for subclasses of sparse graphs. *J. ACM*, 60(5):36:1–36:24, 2013. doi:10.1145/2499483.

23 Zdenek Dvorák and Sergey Norin. Strongly sublinear separators and polynomial expansion. *SIAM J. Discret. Math.*, 30(2):1095–1101, 2016. doi:10.1137/15M1017569.

24 Zdeněk Dvořák and Serguei Norine. Small graph classes and bounded expansion. *J. Comb. Theory, Ser. B*, 100(2):171–175, 2010. doi:10.1016/J.JCTB.2009.06.001.

25 Kord Eickmeyer, Archontia C. Giannopoulou, Stephan Kreutzer, O-joung Kwon, Michal Pilipczuk, Roman Rabinovich, and Sebastian Siebertz. Neighborhood complexity and kernelization for nowhere dense classes of graphs. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017*, volume 80 of *LIPIcs*, pages 63:1–63:14, 2017. doi:10.4230/LIPICS.ICALP.2017.63.

26 Jacob Fox and János Pach. Applications of a new separator theorem for string graphs. *Comb. Probab. Comput.*, 23(1):66–74, 2014. doi:10.1017/S0963548313000412.

27 Delbert Fulkerson and Oliver Gross. Incidence matrices and interval graphs. *Pacific journal of mathematics*, 15(3):835–855, 1965.

28 Cyril Gavoille and Arnaud Labourel. Shorter implicit representation for planar graphs and bounded treewidth graphs. In *15th Annual European Symposium on Algorithms, ESA 2007*, volume 4698 of *LNCS*, pages 582–593. Springer, 2007. doi:10.1007/978-3-540-75520-3_52.

29 Ralucca Gera, Stephen Hedetniemi, and Craig Larson, editors. *Graph theory—favorite conjectures and open problems. 1.* Problem Books in Mathematics. Springer, 2016.

30 Paul W Goldberg, Martin C Golumbic, Haim Kaplan, and Ron Shamir. Four strikes against physical mapping of DNA. *Journal of Computational Biology*, 2(1):139–152, 1995. doi:10.1089/cmb.1995.2.139.

31 Frank Gurski and Egon Wanke. The tree-width of clique-width bounded graphs without $K_{n,n}$. In *Graph-Theoretic Concepts in Computer Science, 26th International Workshop, WG 2000*, volume 1928 of *LNCS*, pages 196–205. Springer, 2000. doi:10.1007/3-540-40064-8_19.

32 Phil Hanlon. Counting interval graphs. *Transactions of the American Mathematical Society*, 272(2):383–426, 1982.

33 Hamed Hatami and Pooya Hatami. The implicit graph conjecture is false. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS 2022)*, pages 1134–1137. IEEE, 2022. doi:10.1109/FOCS54457.2022.00109.

34 David Haussler. Sphere Packing Numbers for Subsets of the Boolean n-Cube with Bounded Vapnik-Chervonenkis Dimension. *J. Comb. Theory, Ser. A*, 69(2):217–232, 1995.

35 Sampath Kannan, Moni Naor, and Steven Rudich. Implicit representation of graphs. In *Proceedings of the twentieth annual ACM symposium on Theory of computing (STOC 1988)*, pages 334–343, 1988. doi:10.1145/62212.62244.

36 Sampath Kannan, Moni Naor, and Steven Rudich. Implicit representation of graphs. *SIAM Journal on Discrete Mathematics*, 5(4):596–603, 1992. doi:10.1137/0405049.

37 Tamás Kővári, Vera T. Sós, and Pál Turán. On a problem of Zarankiewicz. In *Colloquium Mathematicum*, volume 3, pages 50–57. Polska Akademia Nauk, 1954.

38 Ilia Krasikov, Arieh Lev, and Bhalchandra D Thatte. Upper bounds on the automorphism group of a graph. *Discrete Mathematics*, 256(1-2):489–493, 2002. doi:10.1016/S0012-365X(02)00393-X.

39 Daniela Kühn and Deryk Osthus. Induced subdivisions in $K_{s,s}$-free graphs of large average degree. *Comb.*, 24(2):287–304, 2004. doi:10.1007/S00493-004-0017-8.

40 Jiří Matoušek. *Geometric discrepancy*, volume 18 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2010. An illustrated guide, Revised paperback reprint of the 1999 original. doi:10.1007/978-3-642-03942-3.

41 John H. Muller. *Local Structure in Graph Classes*. PhD thesis, Georgia Institute of Technology, 1988.

42 Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-27875-4.

43 Serguei Norine, Paul Seymour, Robin Thomas, and Paul Wollan. Proper minor-closed families are small. *Journal of Combinatorial Theory, Series B*, 96(5):754–757, 2006. doi:10.1016/j.jctb.2006.01.006.

44 Richard Otter. The number of trees. *Ann. of Math. (2)*, 49:583–599, 1948. doi:10.2307/1969046.

45 Felix Reidl, Fernando Sánchez Villaamil, and Konstantinos Stavropoulos. Characterising bounded expansion by neighbourhood complexity. *European Journal of Combinatorics*, 75:152–168, 2019. doi:10.1016/j.ejc.2018.08.001.

46 Jeremy P Spinrad. *Efficient Graph Representations*, volume 19. Fields Institute monographs. American Mathematical Soc., 2003. URL: http://www.ams.org/bookstore-getitem/item=fim-19.

47 William Thomas Tutte. A census of planar triangulations. *Canadian Journal of Mathematics*, 14:21–38, 1962.

48 Emo Welzl. Partition trees for triangle counting and other range searching problems. In Herbert Edelsbrunner, editor, *Proceedings of the Fourth Annual Symposium on Computational Geometry (SoCG 1988)*, pages 23–33. ACM, 1988. doi:10.1145/73393.73397.

49 Édouard Bonnet, Julien Duron, John Sylvester, and Viktor Zamaraev. Adjacency labeling schemes for small classes, 2024. doi:10.48550/arXiv.2409.04821.

A Paths with low crossing number

The main goal of this section is to provide a self-contained proof of Theorem 18. To state the theorem, we recall definitions of some crucial notions. Given a set system $(X, \mathcal{S})$, we say that a set $S \in \mathcal{S}$ *crosses* a 2-element set $\{x, y\} \subset X$ if exactly one of x and y belongs to S. Let $\mathcal{F}$ be a multiset of 2-element subsets in X. The crossing number of $\mathcal{F}$ with respect to a set $S \in \mathcal{S}$ is the number of elements in $\mathcal{F}$ that are crossed by S. The crossing number of $\mathcal{F}$ with respect to $\mathcal{S}$ is the maximum crossing number of $\mathcal{F}$ with respect to a set $S \in \mathcal{S}$. If $(X, \mathcal{F})$, considered as a graph, is a tree or a path spanning all elements of X, then we say that $(X, \mathcal{F})$ is a tree or a path on X, respectively. In this case, the crossing number of $\mathcal{F}$ is referred to as the crossing number of the corresponding tree or path, respectively.

▶ **Theorem 18.** *Let X be an n-element set, $f : \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{\geqslant 0}$ be a strictly increasing function, and d be a natural number. Let $(X, \mathcal{S})$ be a set system of VC dimension d that satisfies $\pi^*_{\mathcal{S}}(m) \leqslant f(m)$ for all $m \in [n]$. Then there exists a path on X with crossing number at most*

$$2 \log |\mathcal{S}| + 10d \cdot \sum_{j=1}^{n} \frac{1}{f^{-1}(j/2)}.$$

This theorem is a known result stated in an unusual form, but one that is useful for us. The main feature of the above formulation is that the bound on the dual shatter function is given in a general form (expressed as a function f) rather than polynomial; in turn, the bound on the crossing number is expressed via the inverse of f. More specifically, the above formulation has two main differences compared to the standard one (for instance [40, 48]). First, in a standard formulation, the bound on the dual shatter function is $\pi^*_{\mathcal{S}}(m) \leqslant m^r$ for some constant $r > 1$. Second, the resulting bound on the crossing number of a path on X is in the form of $O(n^{1-1/r})$. Note that if we take $f(m)$ to be m^r in Theorem 18, then $f^{-1}(m) = m^{1/r}$ and

$$2 \log |\mathcal{S}| + 10d \cdot \sum_{j=1}^{n} \frac{1}{(j/2)^{1/r}}$$

can be upper bounded by $O(n^{1-1/r})$, which recovers the result in a standard form.

The benefit of the present formulation is that one can apply Theorem 18 when a preferable upper bound on $\pi^*_{\mathcal{S}}(m)$ is non-polynomial. For example, if $f(m) = m \cdot g(m)$ for some non-decreasing function $g(m)$, then Theorem 18 gives a bound of $O(g(n) \cdot \log n)$ on the crossing number. In particular, this becomes useful when $g(m) = m^{o(1)}$; e.g., if $g(m) = \log(m)$, the bound on the crossing number becomes $O(\log^2 n)$.

To provide a self-contained proof of Theorem 18, we need to state a number of auxiliary known results in a suitable (non-standard) form and repeat their proofs with some adjustments. Before each statement we explain how its formulation and proof differs from standard ones. We hope this exposition might be of use in some other work.

Packing lemma

We begin with Haussler's Packing Lemma [34] (Lemma 23), which is an improvement over a quantitatively weaker Packing Lemma obtained by Dudley [18] and re-discovered by Welzl [48]. The proof below is a simplification by Chazelle of Haussler's proof [34], which is taken from Matoušek's book [40].

A set system $(X, \mathcal{S})$ is *δ-separated* if for any two distinct sets $S_1, S_2 \in \mathcal{S}$ their symmetric difference contains at least δ elements, i.e., $|(S_1 \setminus S_2) \cup (S_2 \setminus S_1)| \geqslant \delta$. In a standard formulation, the lemma assumes that the shatter function $\pi_{\mathcal{S}}(m)$ is bounded by $O(m^r)$ for some r, and the bound on the cardinality of a δ-separated set $\mathcal{P}$ is $O((n/\delta)^r)$. For our purposes, in Lemma 23, we state the bound on $|\mathcal{P}|$ directly in terms of the shatter function. Note that the standard form of the lemma follows from ours if one uses the bound $\pi_{\mathcal{S}}(m) = O(m^r)$. The difference of the proof below from the one in [40] is that in Equation (13) we bound t in terms of the shatter function directly, rather than in terms of the bound on the shatter function.

▶ **Lemma 23** (Packing Lemma, see [40, Lemma 5.14]). *Let $(X, \mathcal{S})$ be a set system on an n-element set of VC dimension d. Let $\delta \in [n]$ be an integer, and let $\mathcal{P} \subseteq \mathcal{S}$ be δ-separated. Then,*

$$|\mathcal{P}| \leqslant 2 \cdot \pi_{\mathcal{S}}\left(\left\lceil \frac{4dn}{\delta} \right\rceil\right).$$

Before we prove this lemma we must introduce an auxiliary graph construction. For a set system $(X, \mathcal{S})$, we define the *unit distance graph* $\mathrm{UD}(\mathcal{S})$ to be the graph with

$$V(\mathrm{UD}(\mathcal{S})) = \mathcal{S} \qquad \text{and} \qquad E(\mathrm{UD}(\mathcal{S})) = \{\{S, S'\} : |S \Delta S'| = 1\}.$$

We will also need the following bound on the number of its edges.

▶ **Lemma 24** ([40, Lemma 5.15]). *If $(X, \mathcal{S})$ is a set system of VC dimension d on a finite set X then the unit distance graph $\mathrm{UD}(\mathcal{S})$ has at most $d|\mathcal{S}|$ edges.*

With this, we are now ready to prove the Packing Lemma.

Proof of Lemma 23. Choose a random s-element subset $A \subseteq X$, where s is given by

$$s := \left\lceil \frac{4dn}{\delta} \right\rceil. \tag{9}$$

Set $\mathcal{Q} := \{S \cap A \ : \ S \in \mathcal{P}\}$, and for each set $Q \in \mathcal{Q}$ define its weight $w(Q)$ as the number of sets $S \in \mathcal{P}$ with $S \cap A = Q$. Note that

$$\sum_{Q \in \mathcal{Q}} w(Q) = |\mathcal{P}|.$$

Let $E := E(\mathrm{UD}(\mathcal{Q}))$ be the edge set of the unit distance graph $\mathrm{UD}(\mathcal{Q})$, and define an edge weighting of $\mathrm{UD}(\mathcal{Q})$ by giving each edge $e = \{Q, Q'\} \in E$ weight $w(e) := \min(w(Q), w(Q'))$. Let

$$W = \sum_{e \in E} w(e),$$

where we note that this random variable depends on the random choice of A. The desired bound on $|\mathcal{P}|$ in the Packing Lemma is obtained by estimating the expectation of W in two different ways.

First, we claim that for any set $A \subseteq X$, we have the following (deterministic) bound on W

$$W \leqslant 2d \cdot \sum_{Q \in \mathcal{Q}} w(Q) = 2d \cdot |\mathcal{P}|. \tag{10}$$

To see this, note first that the VC dimension of $\mathcal{P}$ is at most d, and thus, by Lemma 24, the unit distance graph $\mathrm{UD}(\mathcal{Q})$ has some vertex Q_0 of degree at most $2d$. By removing Q_0, the total edge weight drops by at most $2d \cdot w(Q_0)$, and we are left with a unit distance graph on $(X, \mathcal{Q}')$, where $\mathcal{Q}' = \mathcal{Q} \setminus Q_0$ and $\mathcal{Q}'$ has VC dimension at most d. We can thus repeat this process until no vertices are left, and we see that the sum of edge weights removed is at most $2d$ times the sum of vertex weights, as claimed.

Next, we bound the expectation $\mathbb{E}[W]$ from below by considering the following random experiment. First, we choose a random $(s-1)$-element set $A \subseteq X$, and then we choose a random element $a \in X \setminus A'$. The set $A = A' \cup \{a\}$ is a uniform random s-element subset of X, we consider $\mathrm{UD}(\mathcal{Q})$ with the same edge weighting as above. Each edge of $\mathrm{UD}(\mathcal{Q})$ is a pair of sets of $\mathcal{Q}$ differing in exactly one element of A. We let $E_1 \subseteq E$ be the edges for which the difference element is a, and let W_1 be the sum of their weights. By symmetry, we have

$$\mathbb{E}[W] = s \cdot \mathbb{E}[W_1]. \tag{11}$$

We are going to bound $\mathbb{E}[W_1]$ from below. Let $A' \subset X$ be an arbitrary but fixed $(s-1)$-element set. We estimate the conditional expectation $\mathbb{E}[W_1 \mid A']$; that is, the expected value of W_1 when conditioned on a fixed set A', and $a \in X \setminus A'$ is selected uniformly at random. Divide the sets of $\mathcal{P}$ into equivalence classes $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_t$ according to their intersection with the set A'. By the definition of $\pi_{\mathcal{S}}(x)$ and since it is non-decreasing, we have

$$t \leqslant \pi_{\mathcal{S}}(s-1) \leqslant \pi_{\mathcal{S}}(s). \tag{12}$$

Let $\mathcal{P}_i$ be one of the equivalence classes, and $b := |\mathcal{P}_i|$. Suppose that an element $a \in X \setminus A'$ has been chosen. If b_1 sets of $\mathcal{P}_i$ contain a and $b_2 = b - b_1$ sets do not contain a, then the class $\mathcal{P}_i$ gives rise to an edge of E_1 of weight $\min(b_1, b_2)$. Note that b_1 and b_2 depend on the choice of a while b does not.

For any non-negative real numbers b_1, b_2 with $b_1 + b_2 = b$, we have $\min(b_1, b_2) \geqslant b_1 b_2 / b$. The value $b_1 b_2$ is the number of ordered pairs of sets (S_1, S_2) with S_1, S_2 being two sets from the class $\mathcal{P}_i$ where one contains a and the other one does not. Now, since $\mathcal{P}$ is δ-separated by hypothesis, if $S_1, S_2 \in \mathcal{P}_i$ are two distinct sets, then they differ in at least δ elements. Therefore, the probability that S_1 and S_2 differ in a random element $a \in X \setminus A'$ is at least $\frac{\delta}{n-s+1} \geqslant \delta/n$. Hence the expected contribution of each pair (S_1, S_2) of distinct sets of $\mathcal{P}_i$ to the quantity $b_1 b_2$ is at least δ/n, thus

$$\mathbb{E}[b_1 b_2 \mid A'] \geqslant b(b-1) \cdot \frac{\delta}{n}.$$

Consequently, the expected contribution $\mathbb{E}[\min(b_1, b_2) \mid A']$ of the equivalence class $\mathcal{P}_i$ to the sum of edge weights W_1 is at least $\mathbb{E}[b_1 b_2/b \mid A'] \geqslant (b-1)\delta/n$. Summing up over all equivalence classes, and using the bound on t from (12),

$$\mathbb{E}[W_1] \geqslant \mathbb{E}[W_1 \mid A'] \geqslant \frac{\delta}{n} \sum_{i=1}^{t} (|\mathcal{P}_i| - 1) \geqslant \frac{\delta}{n} (|\mathcal{P}| - t) \geqslant \frac{\delta}{n} (|\mathcal{P}| - \pi_{\mathcal{S}}(s)). \tag{13}$$

Combining this with the estimate (10), the equality (11), and the definition (9) of s, leads to

$$2d \cdot |\mathcal{P}| \geqslant s \cdot \frac{\delta}{n} \cdot (|\mathcal{P}| - \pi_{\mathcal{S}}(s)) \geqslant 4d \cdot (|\mathcal{P}| - \pi_{\mathcal{S}}(s)).$$

Rearranging gives $|\mathcal{P}| \leqslant 2 \cdot \pi_{\mathcal{S}}(s)$, so the statement follows from (9). ◀

Short Edge Lemma

A standard form of the Short Edge Lemma assumes a polynomial bound on the dual shatter function, i.e., $\pi^*_{\mathcal{S}}(m) = O(m^r)$ for some constant $r > 1$, and the resulting bound is expressed via the bound $O(m^{1/r})$ on the inverse of $\pi^*_{\mathcal{S}}(m)$. In our presentation of the Short Edge Lemma (Lemma 25) below we use a bound on the dual shatter function in a general from expressed as a function f, and the resulting bound is stated in terms of f^{-1}, the inverse of f. As before, by plugging in the bound $\pi^*_{\mathcal{S}}(m) = O(m^r)$ in Lemma 25 one would recover the standard form of the lemma. The proof is taken from [40] with suitable adjustments.

▶ **Lemma 25** (Short Edge Lemma, see [40, Lemma 5.18])**.** *Let X be an n-element set, $f : \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{\geqslant 0}$ be a strictly increasing function, and d be a natural number. Let $(X, \mathcal{S})$ be a set system of VC dimension d with $\pi^*_{\mathcal{S}}(m) \leqslant f(m)$ for all $m \in [n]$. Then, for any multiset $\mathcal{Q}$ with elements from $\mathcal{S}$, there exist elements $x, y \in X$ such that the edge $\{x, y\}$ is crossed by at most*

$$5d \cdot \frac{|\mathcal{Q}|}{f^{-1}(n/2)}$$

sets of $\mathcal{Q}$.

Proof. We form a set system $\mathcal{D}$ which is dual to $\mathcal{Q}$. That is, we consider the multiset $\mathcal{Q}$ as the ground set (if some set appears in $\mathcal{Q}$ several times, then it is considered with the appropriate multiplicity). For each $x \in X$, we let D_x be the set of all sets of $\mathcal{Q}$ containing x, and we set

$$\mathcal{D} = \{D_x : x \in X\}.$$

The symmetric difference $D_x \Delta D_y$ of two sets from $\mathcal{D}$ consists of the sets in $\mathcal{Q}$ crossing the pair $\{x, y\}$. We thus want to show that $D_x \Delta D_y$ is small for some $x \neq y$. We may assume $D_x \neq D_y$ for $x \neq y$, as otherwise we are done, and hence $|\mathcal{D}| = |X| = n$.

The primal shatter function $\pi_{\mathcal{D}}$ is certainly no larger than the dual shatter function $\pi^*_{\mathcal{S}}$, and hence $\pi_{\mathcal{D}}(m) \leqslant f(m)$ by the assumption on $\pi^*_{\mathcal{S}}$. Suppose that any two sets $D_x, D_y \in \mathcal{D}$ have symmetric difference at least δ, then the packing lemma (Lemma 23) implies

$$n = |\mathcal{D}| \leqslant 2 \cdot \pi^*_{\mathcal{S}} \left(\left\lceil \frac{4d|\mathcal{Q}|}{\delta} \right\rceil \right) \leqslant 2f \left(\left\lceil \frac{4d|\mathcal{Q}|}{\delta} \right\rceil \right).$$

Since f is strictly increasing, f^{-1} is also strictly increasing. Therefore, we have

$$f^{-1}(n/2) \leqslant \left\lceil \frac{4d|\mathcal{Q}|}{\delta} \right\rceil \leqslant \frac{5d|\mathcal{Q}|}{\delta},$$

and therefore

$$\delta \leqslant 5d \cdot \frac{|\mathcal{Q}|}{f^{-1}(n/2)},$$

as claimed. ◀

Trees with low crossing number

As with the Short Edge Lemma, the following result (Theorem 26) is stated with a general bound f on the dual shatter function and the resulting bound on the crossing number of a tree is expressed in terms of f^{-1}, the inverse of f. The proof is taken from [40], which is very similar to Welzl's original proof [48]. The statement in [40] is for crossing number of matchings, and therefore it is slightly modified to work for trees as in [48].

▶ **Theorem 26** (see [40, Lemma 5.17]). *Let X be an n-element set, $f : \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{\geqslant 0}$ be a strictly increasing function, and d be a natural number. Let $(X, \mathcal{S})$ be a set system of VC dimension d with $\pi^*_{\mathcal{S}}(m) \leqslant f(m)$ for all $m \in [n]$. Then there exists a tree T on X with crossing number at most*

$$\log|\mathcal{S}| + 5d \cdot \sum_{j=1}^{n} \frac{1}{f^{-1}(j/2)}.$$

Proof. We build T by selecting edges one by one according to the following strategy.

1. Suppose $\{u_1, v_1\}, \ldots, \{u_i, v_i\}$ have already been selected.
2. Define the weight $w_i(S)$ of a set $S \in \mathcal{S}$ as $2^{k_i(S)}$, where $k_i(S)$ is the number of edges among $\{u_1, v_1\}, \ldots, \{u_i, v_i\}$ crossed by S. In particular, $w_0(S) = 1$ for all $S \in \mathcal{S}$.
3. We select the next edge $\{u_{i+1}, v_{i+1}\}$ from $X_i = X \setminus \{u_1, u_2, \ldots, u_i\}$ as a pair of points with the total weight of sets crossing $\{u_{i+1}, v_{i+1}\}$ being the smallest possible.
4. We continue in this manner until $n - 1$ edges have been selected.

Let $E := \{\{u_1, v_1\}, \{u_2, v_2\} \ldots, \{u_{n-1}, v_{n-1}\}\}$. Then $T = (X, E)$ is a tree. Next, we will bound the crossing number k of the resulting tree T. To this end, we estimate the final total weight of all sets of $\mathcal{S}$, i.e.,

$$w_{n-1}(\mathcal{S}) = \sum_{S \in \mathcal{S}} w_{n-1}(S).$$

By definition of w_{n-1} we have $k \leqslant \log w_{n-1}(\mathcal{S})$.

Let us investigate how $w_{i+1}(\mathcal{S})$ increases compared to $w_i(\mathcal{S})$. Let $\mathcal{S}_{i+1}$ denote the collection of the sets of $\mathcal{S}$ crossing $\{u_{i+1}, v_{i+1}\}$. For each set in $\mathcal{S}_{i+1}$ the weight increases by a multiplicative factor of two, and for the other sets it remains unchanged. From this we get

$$w_{i+1}(\mathcal{S}) = w_i(\mathcal{S}) - w_i(\mathcal{S}_{i+1}) + 2w_i(\mathcal{S}_{i+1}) = w_i(\mathcal{S}) \left[1 + \frac{w_i(\mathcal{S}_{i+1})}{w_i(\mathcal{S})}\right].$$

We now estimate $\frac{w_i(\mathcal{S}_{i+1})}{w_i(\mathcal{S})}$ using the Short Edge Lemma (Lemma 25). To begin we define a multiset $\mathcal{Q}_i$ of sets by adding $S \cap X_i$ to $\mathcal{Q}_i$ with multiplicity $w_i(S)$ for every $S \in \mathcal{S}$. Thus

$$|\mathcal{Q}_i| = \sum_{S \in \mathcal{S}} w_i(S) = w_i(\mathcal{S}).$$

Note that, by the construction of $\mathcal{Q}_i$, the number of sets in $\mathcal{Q}_i$ that cross $\{u_{i+1}, v_{i+1}\}$ is $w_i(\mathcal{S}_{i+1})$. Furthermore, by our choice of edge in (2) and (3), the edge $\{u_{i+1}, v_{i+1}\}$ is crossed

by the minimum number of sets in $\mathcal{Q}_i$. Thus, by applying the Short Edge Lemma on the set system $(X_i, \{S \cap X_i \ : \ S \in \mathcal{S}\})$ to the multiset $\mathcal{Q}_i$, and defining $n_i := |X_i| = n - i$, we obtain:

$$w_i(\mathcal{S}_{i+1}) \leqslant 5d \cdot \frac{w_i(\mathcal{S})}{f^{-1}(n_i/2)}.$$

Hence, we have

$$w_{i+1}(\mathcal{S}) \leqslant w_i(\mathcal{S}) \left[1 + \frac{5d}{f^{-1}((n-i)/2)}\right].$$

Therefore

$$w_{n-1} \leqslant w_0(\mathcal{S}) \cdot \prod_{i=0}^{n-1} \left[1 + \frac{5d}{f^{-1}((n-i)/2)}\right] = |\mathcal{S}| \cdot \prod_{j=1}^{n} \left[1 + \frac{5d}{f^{-1}(j/2)}\right].$$

Taking logarithms of both sides and using the inequality $\ln(1+x) \leqslant x$ we obtain

$$k \leqslant \log w_{n-1}(\mathcal{S}) \leqslant \log |\mathcal{S}| + 5d \cdot \sum_{j=1}^{n} \frac{1}{f^{-1}(j/2)},$$

as claimed. ◀

From a tree to a path

Finally, Theorem 18 is obtained from Theorem 26 by using the following Lemma 27 to "convert" a tree with crossing number k to a path with crossing number at most $2k$. The argument is taken from [48] and we provided it here for completeness.

▶ **Lemma 27** ([48, Lemma 3.3])**.** *Let $(X, \mathcal{S})$ be a set system, and T be a tree on X with crossing number at most k with respect to $\mathcal{S}$. Then, there exists a path on X with crossing number at most $2k$ with respect to $\mathcal{S}$.*

Proof. We begin with the following claim.

▷ Claim 28. Given any family $\mathcal{F}$ of two element sets on X, replacing any pair $\{x, y\}, \{y, z\} \in \mathcal{F}$ by the set $\{x, z\}$ will not increase the crossing number with respect to $\mathcal{S}$.

Proof of Claim. If any $S \in \mathcal{S}$ crosses $\{x, z\}$, then exactly one of x or y is not contained in S. Thus, S must cross one of $\{x, y\}$ or $\{y, z\}$. ◁

Let $\mathcal{D} = \{\{u_1, u_2\}, \{u_2, u_3\}, \ldots, \{u_{\ell-1}, u_\ell\}\}$ be a multiset consisting of edges from T in the order they are traversed in a depth-first search (DFS) tour[3] of T starting from an arbitrary vertex. Additionally, let $x_1, \ldots, x_n$ be a labeling of X in the order they are discovered by the DFS tour $\mathcal{D}$.

Note that each edge of T appears exactly twice in $\mathcal{D}$, thus the crossing number of $\mathcal{D}$ is at most $2k$. We can now apply Claim 28 iteratively to reduce $\mathcal{D}$ to $\{\{x_1, x_2\}, \{x_2, x_3\}, \ldots, \{x_{n-1}, x_n\}\}$ without increasing the crossing number with respect to $\mathcal{S}$. ◀

[3] A DFS tour visits all vertices of the tree starting from a given vertex and going as far as possible down a given branch, then backtracking until it finds an unexplored path.

Round-Vs-Resilience Tradeoffs for Binary Feedback Channels

Mark Braverman
Princeton University, NJ, USA

Klim Efremenko
Ben-Gurion University, Beer-Sheva, Israel

Gillat Kol
Princeton University, NJ, USA

Raghuvansh R. Saxena
Tata Institute of Fundamental Research, Mumbai, India

Zhijun Zhang
Princeton University, NJ, USA

Abstract

In a celebrated result from the 60's, Berlekamp showed that feedback can be used to increase the maximum fraction of adversarial noise that can be tolerated by binary error correcting codes from $\frac{1}{4}$ to $\frac{1}{3}$. However, his result relies on the assumption that feedback is "continuous", i.e., after every utilization of the channel, the sender gets the symbol received by the receiver. While this assumption is natural in some settings, in other settings it may be unreasonable or too costly to maintain.

In this work, we initiate the study of *round-restricted feedback channels*, where the number r of feedback rounds is possibly much smaller than the number of utilizations of the channel. *Error correcting codes* for such channels are *protocols* where the sender can ask for feedback at most r times, and, upon a feedback request, it obtains all the symbols received since its last feedback request. We design such error correcting protocols for both the adversarial binary *erasure* channel and for the adversarial binary *corruption* (bit flip) channel. For the erasure channel, we give an exact characterization of the round-vs-resilience tradeoff by designing a (constant rate) protocol with r feedback rounds, for every r, and proving that its noise resilience is optimal.

Designing such error correcting protocols for the *corruption* channel is substantially more involved. We show that obtaining the optimal resilience, even with one feedback round ($r = 1$), requires settling (proving or disproving) a new, seemingly unrelated, "clean" *combinatorial conjecture*, about the maximum cut in weighted graphs versus the "imbalance" of an average cut. Specifically, we prove an upper bound on the optimal resilience (impossibility result), and show that the existence of a matching lower bound (a protocol) is *equivalent* to the correctness of our conjecture.

2012 ACM Subject Classification Theory of computation → Error-correcting codes

Keywords and phrases Round-restricted feedback channel, error correcting code, noise resilience

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.22

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2022/179` [8]

Funding *Mark Braverman*: Supported in part by the NSF Alan T. Waterman Award, Grant No. 1933331, a Packard Fellowship in Science and Engineering, and the Simons Collaboration on Algorithms and Geometry.
Klim Efremenko: Supported by the Israel Science Foundation (ISF) through grant No. 1456/18 and European Research Council Grant number: 949707.
Gillat Kol: Supported by a National Science Foundation CAREER award CCF-1750443 and by a BSF grant No. 2018325.
Raghuvansh R. Saxena: Supported by the Department of Atomic Energy, Government of India, under project no. RTI4001.
Zhijun Zhang: Supported by a National Science Foundation CAREER award CCF-1750443.

Acknowledgements We thank Noga Alon for discussions regarding Conjecture 3.



© Mark Braverman, Klim Efremenko, Gillat Kol, Raghuvansh R. Saxena, and Zhijun Zhang;
licensed under Creative Commons License CC-BY 4.0



1 Introduction

Cybernetics. Consider the following two scenarios. Scenario one: a steersperson wishes to steer a longship to shore. She maintains a steady course in a changing environment (wind, waves, storms, currents, tides, etc.) by adjusting her steering in continual response to the effect it is observed as having. Scenario two: a teacher has a semester-worth of topics he wishes to teach to his class. He schedules exams throughout the semester to help him adapt his pace and determine what material should be repeated.

The above two scenarios are examples of *cybernetics*, a field that studies *self-regulating* processes. A core concept in cybernetics is *circular causality*, which is typically implemented using *feedback* mechanisms, where the observed outcomes of actions are taken as inputs for further actions. This is the case for, e.g., spacecraft navigators, artificial limbs, and our bodies' regulation of hormone and blood sugar levels. The term Cybernetics[1] was coined in 1948 by the mathematician and philosopher Norbert Wiener for "the science of control and communication in the animal and the machine" [37], following exchanges between numerous fields during the 1940s, including anthropology, mathematics, neuroscience, psychology, and engineering.

Feedback in information theory. Cybernetics grew alongside and built on Claude Shannon's information theory, that was developed to improve the transmission of information and introduced the notion of *error correcting codes*. Shannon was interested in knowing whether the existence of a "feedback link" in the channel, where after every utilization of the channel, the (possibly incorrect) symbol obtained by the receiver is also given to the sender, allows for better codes. A discouraging early result by Shannon showed that feedback does not improve the capacity of memoryless channels [31]. It would be another decade or so before Berlekamp proves that feedback can, in fact, increase the maximum fraction of *adversarial* errors that can be tolerated. Specifically, Berlekamp showed that the *maximum noise resilience* of the (adversarial) binary channel increases from $\frac{1}{4}$ to $\frac{1}{3}$ given feedback [4, 5] (also see [39, 35, 1]).

A key property of the feedback channel exploited by Berlekamp's result, as well as by follow up work, is that it supports "*continuous*" feedback – after *every* communication round, the sender gets the symbol received by the receiver. This assumption is natural in some settings, e.g., in scenario one, the steersperson continuously watches the ship's motion as she steers. However, this assumption may be unreasonable or too costly to maintain in other settings, e.g., in scenario two, the teacher may not want to continuously quiz his students.

This work: round-restricted feedback. Motivated by such examples, in this work, we initiate the study of *round-restricted feedback channels*, where the number of feedback rounds is possibly much smaller than the number of communication rounds. Specifically, we wish to design protocols with optimal noise resilience that allow the sender (Alice) to transmit a message to the receiver (Bob), where during the execution of the protocol, the sender can ask for feedback at most r times. Upon such a request, the sender obtains all the bits received by the receiver from the last time feedback was solicited.

One can consider two models for scheduling the feedback rounds: the *adaptive* and the *non-adaptive* models. In the non-adaptive model, the sender decides ahead of time (before the protocol is run and before the input is known) when to schedule the r feedback rounds, while in the adaptive model, the timing of each feedback request may depend on the previously received feedback. In the second scenario, for example, the non-adaptive setting

[1] Interestingly, Cybernetics comes from the Greek word "Kubernetes", which means steersperson.

corresponds to scheduling all exams at the beginning of the semester, while the adaptive setting corresponds to scheduling the next exam after the previous one was given. While our techniques hold for both the adaptive and non-adaptive settings, we choose to present our results for the non-adaptive setting. See Section 1.3 and Section 2.1 for a discussion of the implication of our techniques for the adaptive setting.

We consider such message transmission protocols with r feedback rounds over both the (adversarial) binary *erasure channel*, that erases some of the sent bits (those bits are received as "$\perp$"), and over the (adversarial) binary *corruption channel*, that flips some of the sent bits. As was mentioned before, classical results in information theory show that with no feedback the maximum noise resilience of the binary corruption channel is $\frac{1}{4}$ [25], while with continuous feedback, the maximum resilience improves to $\frac{1}{3}$ [4, 5]. For the binary erasure channel, it is known that with no feedback the maximum resilience is $\frac{1}{2}$, and it is easy to see that with continuous feedback it approaches 1: the sender re-transmits each symbol until the receiver receives it.

We mention that rounds (or passes) are often considered to be a scarce resource and that round-restricted algorithms are extensively studied in other communication settings, e.g., communication complexity, distributed computing, streaming algorithms, and cryptographic protocols, and that we draw inspiration from these settings.

1.1 Our Results and Conjecture

Due to page limit, sections containing formal discussion on the corruption case and all proofs are deferred to the full version [8]. Here, we focus on the erasure case and only provide high level ideas for the corruption case.

1.1.1 The (Adversarial) Binary Erasure Channel

As discussed above, the maximum resilience of the erasure channel is known for the extreme cases of no feedback and of continuous feedback. Our first result is an optimal *round-vs-resilience tradeoff* for the erasure channel with any number of non-adaptive feedback rounds.

▶ **Theorem 1.** *The maximum noise resilience of the (adversarial) binary erasure channel with r rounds of feedback is $\frac{5}{7}$ if $r = 1$ and $1 - \frac{7}{12(r+1)}$ if $r > 1$. Furthermore, the maximum noise resilience can be obtained by a* deterministic, constant-rate *protocol.*

Theorem 1 can be viewed as a "hierarchy theorem", showing that more feedback rounds allow for strictly better resilience. On the other hand, Theorem 1 also shows that a constant number $O_\epsilon(1)$ of feedback rounds already suffices to get a noise resilience of $1 - \epsilon$ for the erasure channel.

Techniques. The main ingredient in our proof of Theorem 1 is the construction of a *list decodable* code for the binary erasure channel with m codewords, for *all* (not necessarily asymptotic) values of m. Our code is optimal in the sense that it achieves the maximum error resilience for every list size simultaneously. We emphasize that for our protocols, we need such a code for all possible m, which corresponds to all possible "block sizes". We call codes with small m's "*small codes*". Given these codes, the protocols we use to prove Theorem 1 are rather simple – after every feedback round, Alice and Bob agree on a (smaller, unless there was a lot of noise) set Γ of candidate inputs x and Alice encodes x with our optimal list decodable code with $m = |\Gamma|$ codewords. On the analysis front, we are able to

argue that, unless the adversary erases many of the sent bits, the size of the candidate set Γ shrinks substantially between feedback rounds, and measure this shrinkage exactly. See Section 2.1 for a detailed overview.

1.1.2 The (Adversarial) Binary Corruption Channel

Theorem 1 gives a complete characterization of the noise resilience of the erasure feedback channel as a function of the number of feedback rounds. However, as will be explained next, the case of corruptions is much more involved, and we will focus on protocols with one round of feedback. We mention that since the adaptive and non-adaptive models are the same for protocols with one feedback round, the results in this section hold for both the adaptive and non-adaptive settings. Our next theorem gives an upper bound on the noise resilience of such one-round protocols.

▶ **Theorem 2.** *The maximum noise resilience of the (adversarial) binary corruption channel with one round of feedback is at most* $\frac{7}{23}$.

We conjecture that the upper bound of $\frac{7}{23}$ on the noise resilience in Theorem 2 is tight, and that it can be achieved by a constant-rate protocol. Perhaps surprisingly, proving this is *equivalent* to showing the following combinatorial conjecture about the existence of large cuts in graphs.

▶ **Conjecture 3.** *Let G be a graph with n vertices and non-negative edge weights summing up to 1. Let $\mathsf{wt}(S)$ be the sum of weights of all the edges with both endpoints in the subset of vertices S, and let $\mathsf{Max\text{-}Cut}(G)$ be the maximum total weight of all the edges across any cut in G. Then,*[2]

$$\mathsf{Max\text{-}Cut}(G) \geq \tfrac{2}{3} - \tfrac{16}{15} \cdot \mathop{\mathbb{E}}_{S\subseteq[n]}\left[\min\left(\mathsf{wt}(S), \mathsf{wt}(\overline{S})\right)\right]. \tag{1}$$

We prove Conjecture 3 for (large enough) graphs where all edges have equal weight, i.e., "unweighted" graphs. However, the case for general weighted graphs seems much harder, and, despite our best effort, we were unable to prove (or disprove) it. We also mention that Conjecture 3 is tight for some graphs (e.g., cliques of size 3 and 5 with edges of equal weight), and related bounds on $\mathsf{Max\text{-}Cut}$ were studied in other contexts, e.g., [26, 2, 22].

The next theorem gives the equivalence between Conjecture 3 and the tightness of Theorem 2.

▶ **Theorem 4.** *Theorem 2 is tight if and only if Conjecture 3 holds. Furthermore, Conjecture 3 implies a* constant rate *protocol achieving the maximum noise resilience.*

In essence, Theorem 4 connects the problem of designing optimal error correcting protocols with one round of feedback to a combinatorial question about graphs. As we discuss later in Section 2.2, our techniques can also be used to connect the problem of designing optimal error correcting protocols with multiple rounds of feedback to similar questions about graphs.

---

[2] As the expectations of $\mathsf{wt}(S)$ and $\mathsf{wt}(\overline{S})$, for a uniformly random S, are $\frac{1}{4}$, Equation (1) can be equivalently written as $\mathsf{Max\text{-}Cut}(G) \geq \frac{6}{15} + \frac{8}{15} \cdot \mathbb{E}_{S\subseteq[n]}\left[\left|\mathsf{wt}(S) - \mathsf{wt}(\overline{S})\right|\right]$, where the term inside the expectation is the "imbalance" of a random cut.

Techniques. The proof of Theorem 4 is technically involved and a detailed overview can be found in Section 2.2. At a high level, the main ingredient in designing our protocol is the construction of a special type of "weighted" codes, called dc-*codes*. A dc-code C is parameterized by a "distance contribution function" dc that assigns a value in $[0,1]$ to each possible message $x \in \{0,1\}^k$. We require that for all $x \neq x' \in \{0,1\}^k$, the codewords $C(x)$ and $C(x')$ are at least (relative) Hamming distance $\mathsf{dc}(x) + \mathsf{dc}(x')$ apart. Equivalently, we ask that the balls of radii $\mathsf{dc}(x)$ around $C(x)$ are all disjoint.[3] We note that unlike traditional error correcting codes that have only one distance guarantee for all pairs of codewords (i.e., the minimum distance), the distance guarantees for different pairs of codewords in a dc-code are different. In fact, traditional codes can be viewed as dc-codes for a constant dc function.

dc-codes for non-constant dc functions are useful for our protocol as if the adversary already used up many of its corruptions before the feedback round, Alice knows she can afford to send her message x encoded with an error correcting code that does not guarantee a large distance between $C(x)$ and the other codewords. Geometrically, designing a dc-code is a sphere packing problem where we need to pack spheres of different radii $\mathsf{dc}(x)$. As for some x's a small radius $\mathsf{dc}(x)$ suffices, some of the spheres are small, which allows the other spheres being packed to be larger.

The proof of Theorem 4 shows that Conjecture 3 implies the existence of dc-codes that are needed for our protocol to work. We assume that Alice uses a uniformly random code to encode her message before the feedback. The codeword sent by Alice can be corrupted by the channel in many ways, and each such way would imply a function dc such that Alice would like to use a dc-code to encode her message after the feedback. We denote by Q the set of dc functions for which the corresponding dc-codes are needed by our protocol. We also denote by P the set of dc functions for which dc-codes exist. We wish to show $Q \subseteq P$. To this end, we show that both P and Q are closed and convex, and that in every direction z, the extremal point of P in direction z is "farther" than the extremal point of Q in direction z. We then recast this geometric problem as a combinatorial problem by interpreting the direction vector z as a weighted graph G, and show that the extremal point of P in direction z corresponds to a Max-Cut in G (as in the left hand side of Conjecture 3), while the extremal point of Q in direction z corresponds to the right hand side of Conjecture 3.

For the converse direction of Theorem 4, we show that the arguments in the above paragraph are actually equivalences, except for the assumption that Alice uses a randomly sampled code to encode her message before the feedback. At a high level, we use *Ramsey theory* to show that the assumption that this code is a random code is, at least in some sense, without loss of generality (see Section 2.2.2 for a more precise statement).

1.2 Related Work

Feedback channels were studied since the early days of information theory and are still actively studied [31, 24, 14, 5, 9, 27, 32, 13, 33, 34, to cite a few]. While feedback does not increase the capacity of discrete memoryless channels with vanishing error, there are settings where feedback is known to allow improvement, like in the 0-error capacity case [31], and under variable decision time [9].

Partial feedback. Haeupler, Kamath, and Velingker [23] considered the setting where the feedback is partial, and showed that even if Alice receives feedback bits from Bob for an arbitrarily small constant fraction of her transmissions, resilience close to (the optimal

[3] We mention that dc-codes are an example of non-equally spaced codes defined in [11].

resilience of) $\frac{1}{3}$ is possible using a randomized protocol. However, the number of feedback rounds their protocol needs grows linearly with n, the length of Alice's input. See [36] for a subsequent result.

Independently and concurrently to our work, [15] improved [23] and showed a deterministic protocol that uses $\mathcal{O}(\log n)$ feedback bits over $\mathcal{O}(1)$ feedback rounds to get resilience approaching $\frac{1}{3}$, along with a similar result for the erasure channel showing that the resilience approaches 1 for this channel. The main difference between [15] and the current work is that we focus on finding the optimal resilience for any given number r of feedback rounds whereas [15] focuses on showing that the resilience approaches the optimal value as the constant r increases. Additionally, their work measures both the number of feedback rounds and the number of feedback bits, while we only focus on the number of rounds.

Two-way codes and interactive codes. As discussed above, feedback is also known to increase the noise resilience of the adversarial binary corruption channel [4, 5], and this result played a big role in recent work in *interactive coding* [10, 18, 17] and *two-way coding* [16, 19, 11]. In interactive coding [28, 29, 30], we wish to simulate a communication protocol Π that was designed to work over the noiseless channel, by a protocol Π' that works over a noisy channel. In the setting of two-way codes, like in the setting of traditional error correcting codes, Alice wishes to transmit a message x to Bob over a noisy channel. However, unlike the case of traditional codes, where Alice is the only party that can transmit messages, in two-way codes Bob can also use the (noisy) channel to transmit messages back to Alice.

Observe that since Bob has no input, any two-way code can be run over the feedback channel and thus two-way error correcting codes can be viewed as protocols over a *noisy feedback* channel. In particular, since the noise tolerance of the binary corruption channel is only $\frac{1}{3}$, the noise resilience of binary two-way codes over the binary corruption channel is at most $\frac{1}{3}$. In the same way, results for the bounded round feedback channel give upper bounds on the noise resilience of the corresponding two-way channels.

Gupta, Kalai, and Zhang [16, 19] studied two-way error correcting codes over the binary *erasure channel*. Their main result is a code that is resilient to a $\frac{3}{5}$ fraction of adversarial errors, improving on the noise tolerance of the one-way binary erasure channel that is known to be $\frac{1}{2}$. We mention that the two-way coding schemes of [16, 19] exchange (almost) linear number of messages. The work of [16] also gives an upper bound of $\frac{2}{3}$ on the maximum tolerance of the two-way binary erasure channel, and an upper bound of $\frac{2}{7}$ on the maximum tolerance of the two-way binary corruption channel. Given those upper bounds, a corollary of our results is that even a single round of noiseless feedback allows for a better error tolerance than any number of noisy feedback rounds over both the erasure and corruption channels[4].

The recent work of Efremenko, Kol, Saxena, and Zhang [11] shows that the maximum noise resilience of two-way error correcting codes for the binary corruption channel is strictly better than the noise resilience of traditional error correcting codes for this channel, which is

[4] To see why, observe that if Bob's messages are noiseless, we can assume without loss of generality that Bob's messages are much shorter, say at most an ϵ fraction, of Alice's messages. Indeed, if not, consider a modified protocol where all messages from Alice are repeated k times, for some large k. For the erasure channel, either all the repetitions of a bit from Alice are erased or Bob knows the bit exactly. Thus, his communication does not grow with k. For the corruption channel, it suffices for Bob to say how many of the repetitions were received as 1, which can be done using $\log k \ll k$ bits.
Moreover, we mention that for this claim, we do not need to rely on Conjecture 3, as a lower bound slightly smaller than $\frac{7}{23}$ (but greater than $\frac{2}{7}$) on the maximum error resilience of protocols with one feedback round over the binary corruption channel can be obtained unconditionally using our techniques (but is not included in the current work).

known to be $\frac{1}{4}$ [25]. At a very high level, those results for two-way codes are obtained by implementing a (weak) feedback mechanism over channels with no built-in feedback. Related ideas were used in [10, 18, 17] to give interactive binary error correcting codes with high noise resilience.

List decodable codes. List decodable codes were introduced in the 50's [12, 38] and have been studied over numerous papers and found many applications since then. We next list the works most related to ours. Most of the work on list decoding was done in the asymptotic regime, where the number of codewords goes to infinity. In this work, we are interested in the optimal list decodable codes for any (potentially small) number of codewords. However, as an ingredient in our proof, we use the asymptotic results of [3] (see also [6, 21, 7]) for optimal list decoding of the corruption channel (see Lemma 11). The list decoding question was also considered for other channels, for example, over the corruption channel with feedback [32] and the erasure channel [20].

1.3 Open Problems

Our work suggests the study of feedback channels through a new lens, namely, their feedback round complexity. We next list some suggestions for future work in this direction.

Graph-theoretic conjectures. The most immediate question we leave open is proving Conjecture 3 for all weighted graphs. We also propose the following potentially related conjecture, which is tight for all odd cliques with edges of equal weight.

▶ **Conjecture 5.** *Let G be a graph with n vertices and non-negative edge weights summing up to 1. Let $\mathsf{wt}(i)$ be the sum of weights of all edges incident on vertex i. Then,*

$$\mathsf{Max\text{-}Cut}(G) \geq \frac{1}{2} + \frac{1}{8} \cdot \sum_{i \in [n]} \mathsf{wt}(i)^2.$$

Round-vs-resilience tradeoff for other channels. Proving Conjecture 3 would imply that our protocol in Theorem 4 has optimal noise resilience among protocols with one round of feedback over the corruption channel. Obtaining a general round-vs-resilience tradeoff for any number of feedback rounds r for the corruption channel and for other well-studied channels (e.g., the binary insertion-deletion channel[5], the binary deletion-only channel, and non-binary channels), would be interesting.

Adaptive corruptions over the erasure channel. Theorem 1 considers the case of *non-adaptive* feedback rounds, where Alice decides ahead of time when to ask for feedback. It can be shown that the case of *adaptive* feedback rounds, where Alice chooses when to ask for another round of feedback after seeing the previous feedback, allows for (strictly) better round-vs-resilience tradeoffs. Our techniques can be used to write a recursive formula for the noise resilience in the adaptive case, and finding a "clean", closed-form formula for this setting (if one exists) is left open (see Section 2.1).

[5] We note that this first requires a suitable definition for the insertion-deletion channel with constant number of rounds.

2 Proof Overview

In this section, we overview the proofs of Theorems 1 and 4, starting with the relatively easier Theorem 1.

2.1 Result for the Erasure Channel – Theorem 1

The defining feature of the erasure channel is that the receiver (Bob) either receives the bit sent by Alice or receives a special erasure symbol $\perp$. This means that in any round where Bob receives $\perp$, he is certain that this is due to the erasures in the channel, while if he receives a symbol different from $\perp$, he is certain that the symbol must be what Alice sent in that round. In turn, this means that Bob knows exactly the amount of erasures introduced by the channel and also means that Bob can (recall that he is trying to determine Alice's input) remove from consideration any candidate input that is "inconsistent" and would make Alice send a different symbol in any such round.

The general format of a protocol. The above observation implies that protocols for the erasure channel with r rounds of feedback (and therefore $r+1$ messages from Alice) have the following format: Alice starts with an input $x \in \Gamma_0 = \{0,1\}^n$. For her first message, she takes a code[6] $C_0 : \Gamma_0 \to \{0,1\}^*$ and sends $C_0(x)$ to Bob. Some of the bits of $C_0(x)$ are received correctly by Bob, while the remaining bits are erased and replaced with $\perp$. Using the bits he received correctly, Bob can calculate the number of erasures N_1 introduced by the channel in this round and can identify a subset $\Gamma_1 \subseteq \Gamma_0$ of inputs for Alice that are consistent with the message he received. Note that Alice's input x must be in Γ_1.

Then, a feedback round takes place, and as Alice learns all the received symbols, she can also compute N_1 and Γ_1. As both parties now know these values, they can now "forget" this round and "reduce"[7] to a smaller problem where Alice wants to transmit an element $x \in \Gamma_1$ to Bob using a protocol with $r-1$ rounds of feedback and the maximum number of erasures the channel can insert is N_1 lower than what it was before. Continuing this way, the goal of the parties is to reduce to a problem with 0 rounds of feedback, and set of inputs Γ_r such that there exists a (standard) error correcting code for elements in Γ_r resilient to the number of erasures that the channel can insert in the last round.

List-decodable small codes. It is readily seen that the protocol format described above does not care about the exact strings in the sets $\Gamma_0, \ldots, \Gamma_r$, as long as their sizes stay the same. Thus, the question of whether or not the above protocol format can be instantiated to get a protocol that is resilient to θ fraction of adversarial erasures, for some $\theta \in [0,1]$, reduces to determining when to schedule the feedback rounds, and given two feedback rounds, determining the codes C_i to be used by Alice between these rounds. The codes C_i should be such that, given an initial set size $m = |\Gamma_i|$ and a target set size[8] $k = |\Gamma_{i+1}|$, the number

[6] At this point, it may be helpful to view this as a function instead of a code. We explain why we are calling it a code later. Also, a more precise way to state this would be to say that there exists an $L > 0$ such that $C_0 : \Gamma_0 \to \{0,1\}^L$, as all codewords need to be of the same length to avoid the parties from signaling through the length of the codeword. Nonetheless, we stick with statements like $C_0 : \Gamma_0 \to \{0,1\}^*$ throughout this sketch for simplicity.

[7] We elaborate what this means exactly in the paragraph on adaptive feedback rounds below.

[8] Note that k is not known to the parties in advance, and thus it will be ideal if the code used is optimal for all k simultaneously.

of erasures required to reduce the set size from m to k is the highest. Using such codes, Alice ensures that unless the adversary invests many erasures, the set of candidates shrinks substantially between feedback rounds. We first focus on designing such codes.

Codes like the above are known as list decodable codes, and have been well studied in the asymptotic regime, where m tends to infinity, and exact answers are known (see, e.g., [12, 38, 6, 20, 7, 3, 32] and Lemma 11). However, for our purposes, we need the exact answer for smaller values of m as well. Codes with small m, i.e., "small codes" or codes with few codewords, have recently received a lot of attention and have proven to be useful in designing binary protocols with high error resilience in several contexts [10, 11, 16, 18, 19]. In the current paper, we provide a complete analysis of the list-decodability of these codes for the erasure channel, giving a function $\mathsf{d}(m,k)$ that characterizes exactly the minimum amount of erasure noise needed such that for *any* code $C : [m] \to \{0,1\}^*$, one can erase $\mathsf{d}(m,k)$ fraction of the bits and ensure that Bob gets a list of candidates of length strictly smaller than k.

The formula for $\mathsf{d}(m,k)$ is given in Equation (7). Proving that this formula is correct requires showing both a construction (of codes with resilience approaching $\mathsf{d}(m,k)$) and an impossibility result. Our construction has the nice property that the same code is tight simultaneously for all values of k. Roughly speaking, our code achieves this optimal erasure noise resilience by ensuring that every coordinate is as differentiating as possible, i.e., we ensure that for all coordinates j, exactly $\lfloor \frac{m}{2} \rfloor$ (uniformly chosen) codewords have 0 in that coordinate, while the remaining $\lceil \frac{m}{2} \rceil$ codewords have 1 (see Lemmas 9 and 10). This is as opposed to randomly sampled codes where, e.g., a $\frac{1}{2^m}$ fraction (which is large for small m) of the coordinates are expected to be 0 for all the codewords, and therefore not differentiate between any pair of codewords.

Scheduling the feedback rounds. Even with an exact formula for $\mathsf{d}(m,k)$ in hand, it still remains to schedule the feedback round correctly in order to maximize the overall noise resilience of the obtained protocol. The fact that our constructed code is tight simultaneously for all values of k is of great help for this part, as the actual value of k is determined by the erasures inserted by the channel and not in our control. This means that in order to schedule the feedback rounds optimally, one needs to go over all possible values of k (across all rounds) that may happen over the channel and maximize the corresponding error resilience. This requires a careful analysis of the obtained formula for $\mathsf{d}(m,k)$.

Adaptive feedback rounds. We finish this section by briefly discussing the extension of our result to adaptive feedback rounds, as hinted in Section 1.3. Recall our reduction above from r to $r-1$ feedback rounds, and note that this reduction is not perfect in the following sense: the erasures inserted by the adversary in Alice's first message in the r-round protocol dictate the set Γ_1 of candidates and the budget of the $(r-1)$-round protocol. Observe that the $(r-1)$-round protocol with maximal noise resilience for transmitting a message depends on the size of the set of candidates and on the erasure budget. Now, since our r-round protocol is non-adaptive, meaning that the timing of all feedback rounds is fixed in advance and cannot be recalculated given the erasures in the first round, our r-round protocol may reduce to a sub-optimal $(r-1)$-round protocol. Therefore, when scheduling the feedback rounds for our r-round protocol, one needs to consider the values of k that are possible across all rounds in order to get the optimal schedule.

On the other hand, if the feedback rounds can be scheduled adaptively, the reduction is indeed perfect. In this case, one just needs to schedule the first feedback round beforehand based on the possible values of $k = |\Gamma_1|$ for this round alone, and then, upon seeing the N_1

and Γ_1 values, one can take the $(r-1)$-feedback round protocol with the maximum error resilience (when Alice's input is from Γ_1 and the number of erasures is N_1 lower) and schedule the remaining feedback rounds according to this protocol. Thus, our techniques also lead to a tight recursive formula for the maximum error resilience in the case of adaptive feedback rounds, but converting it to a "clean" closed form formula (if at all possible) is left open.

2.2 Result for the Corruption Channel – Theorem 4

Compared to the erasure channel, where Bob knows exactly the amount of noise inserted and can safely eliminate many candidate inputs for Alice, the corruption channel is much harder. Here, upon receiving a message from Alice, all Bob can compute is, given a candidate input y for Alice, what is the number $\mathsf{N}(y)$ of corruptions the channel inserted assuming Alice's input was indeed y. Crucially, this value of $\mathsf{N}(y)$ may be very different for different y, and unless it exceeds the maximum possible number of corruptions in the channel (which can only happen when the protocol is quite far advanced), it can never have Bob eliminate y from consideration entirely.

Consider now a protocol over the corruption channel with one round of feedback (and therefore, two messages from Alice). Suppose that Alice's input x comes from the set $\{0,1\}^n$. As explained above, after receiving the first message from Alice, Bob knows $\mathsf{N}(y)$ for all $y \in \{0,1\}^n$. By subtracting $\mathsf{N}(y)$ from the maximum possible number of corruptions, Bob can compute, for all $y \in \{0,1\}^n$, a number $\mathsf{dc}(y)$ which is the leftover corruptions, or, equivalently, the degree to which the second message of Alice can be corrupted, assuming her input is y. As Alice receives feedback from Bob, she can also compute the values $\mathsf{dc}(y)$ for all $y \in \{0,1\}^n$. In the remainder of this sketch, we normalize $\mathsf{dc}(y)$ by dividing it by the length of Alice's second message. This will result in a value in $[0,1]$.

dc-codes. Using this feedback, Alice's goal in her second message is to allow Bob to uniquely identify her input. If $C : \{0,1\}^n \to \{0,1\}^*$ is the code used by Alice in her second message, the only way Bob can uniquely decode Alice's input is if for all $y \neq y' \in \{0,1\}^n$, the codewords $C(y)$ and $C(y')$ are at least (relative) Hamming distance $\mathsf{dc}(y) + \mathsf{dc}(y')$ apart. The reason is that if y is Alice's input, then the adversary has fractional budget $\mathsf{dc}(y)$ that it can use to corrupt $C(y)$, and thus the codeword received by Bob can be any string of (relative) Hamming distance at most $\mathsf{dc}(y)$ from $C(y)$. Similarly, if y' is Alice's input, then the codeword received by Bob can be any string of Hamming distance at most $\mathsf{dc}(y')$ from $C(y')$. Note that the adversary cannot arrange for the received encodings to be the same if and only if $C(y)$ and $C(y')$ are at least (relative) Hamming distance $\mathsf{dc}(y) + \mathsf{dc}(y')$ apart. We call a code that satisfies this (relative) Hamming distance property a dc-code and mention that the values $\mathsf{dc}(y)$ can equivalently be seen as the "distance contributed" by y in such a code.

We note that unlike traditional error correcting codes that have only one distance guarantee for all pairs of codewords (i.e., the minimum distance), for dc-codes, the distance between a pair of codewords may be different depending on the "compatibility" of the messages they encode. Specifically, we think of each codeword as having a different "radius" and the code needs to "pack" all the induced balls of different radii. We point out that dc-codes are an example of non-equally spaced codes defined in [11].

We also observe that the small code used in our protocol for erasures can be viewed as a dc-code where $\mathsf{dc}(y) = 0$ for all inputs y that Bob has ruled out (and therefore, do not need any distance guarantees), and $\mathsf{dc}(y) = c$ for all inputs y that he has not ruled out, where c is

the best possible constant (c is determined by the $\mathsf{d}(m,k)$ function). We mention that for the erasure channel, our protocol also needed list-decoding guarantees that are not needed here as we are only attempting to get a one feedback round protocol.

The discussion so far shows that the existence of a protocol with a given error resilience amounts to determining whether or not it holds that for all functions $\mathsf{dc}(\cdot)$ that can be induced by the corruptions inserted in Alice's first message, there exists a dc-code that Alice can use to compute her second message. Curiously, we show in the next subsection that this question is equivalent to our seemingly unrelated combinatorial conjecture (Conjecture 3) about the existence of large cuts in graphs.

Towards multiple rounds of feedback. The above approach of designing dc-codes (that have no rounds of feedback) to construct protocols with one round of feedback can be generalized. One can similarly argue that, for any $r \geq 0$, dc-codes with r rounds of feedback can be used to construct protocols with $r+1$ rounds of feedback. Analogously to the above, the "extra" round is the first round, and dictates which dc-code is used in the rest of the protocol. Moreover, questions about constructing dc-codes with r rounds of feedback can be translated to questions about graphs. The $r = 0$ case is explained next, but similar ideas may be used for general r, with appropriate changes in the definitions of the set P and Q (see below).

2.2.1 Conjecture 3 Implies a Tight Protocol

We first show why Conjecture 3 implies the existence of a tight protocol. In fact, we shall show the existence of a protocol where Alice's message in the first round is simply the encoding of her input x using a randomly sampled code. Let $m = 2^n$. A distance function is a function $\mathsf{dist} : \binom{[m]}{2} \to \mathbb{R}$, where $\binom{[m]}{2}$ is the set of all subsets of $[m]$ of size 2. For a code $C : [m] \to \{0,1\}^*$, we denote by dist_C the distance function induced by C, i.e., $\mathsf{dist}_C(i,i')$ is the (relative) Hamming distance between $C(i)$ and $C(i')$. For a distance contribution function dc, we denote by $\mathsf{dist}_{\mathsf{dc}}$ the distance function induced by dc, i.e., $\mathsf{dist}_{\mathsf{dc}}(i,i') = \mathsf{dc}(i) + \mathsf{dc}(i')$. For simplicity, throughout this overview we assume that $\mathsf{dc}(y) = 1 - \mathsf{N}(y)$ (recall that $\mathsf{dc}(y)$ is actually the normalized leftover corruption count, but in this sketch we will ignore the exact multiplicative and additive constants in this function).

Recasting as a geometric problem. We denote by P the set of all distance functions dist_C that are induced by codes $C : [m] \to \{0,1\}^*$. We denote by Q the set of all distance functions $\mathsf{dist}_{\mathsf{dc}}$ induced by dc functions that can be obtained by the corruptions inserted in Alice's first message (recall that dc depends on N, which is a function of the corruptions inserted in Alice's first message). In other words, P is the set of distance functions that can be realized and Q is the set of distance functions required by our protocol. We wish to prove $Q \subseteq P$.

We view distance functions dist as $\binom{m}{2}$-dimensional vectors. We observe that both P and Q are closed and convex and that the set P is "downwards-closed", meaning that if $\mathsf{dist} \in P$ then any dist' that is coordinate wise at most dist is also in P. This means that showing $Q \subseteq P$ is equivalent to showing that for all $\binom{m}{2}$-dimensional non-negative hyperplanes z, it holds that:

$$\max_{\mathsf{dist} \in P} \langle z, \mathsf{dist} \rangle \geq \max_{\mathsf{dist} \in Q} \langle z, \mathsf{dist} \rangle, \tag{2}$$

Recasting as a combinatorial problem. By scaling, we can assume that the entries of z sum to 1 and view them as the weights on the edges of an m-vertex graph G_z as in Conjecture 3. As both P and Q are closed and convex, both the maximums are attained at one of their vertices.

To reason about Equation (2), it will be useful to represent a code $C : [m] \to \{0,1\}^L$ as a sequence of L one-bit functions $b : [m] \to \{0,1\}$ (the first one-bit function corresponds to the first coordinate of $C(i)$, etc.). Observe that for the code $b : [m] \to \{0,1\}$ (i.e., $L = 1$), it holds that dist_b is a boolean function with $\mathsf{dist}_b(i,i') = 1$ if and only if $b(i) \neq b(i')$.

The LHS of Equation (2). Since a general code C is a sequence of one-bit functions, it can be shown that the function dist_C is a convex combination of the functions dist_b that are induced by one-bit functions b. In particular, this means that the vertices of P are distance functions induced by one-bit functions. Using the expression above for dist_b for one-bit function $b : [m] \to \{0,1\}$, the value of $\langle z, \mathsf{dist}_b \rangle$ is the value of the cut in the graph G_z indicated by b:

$$\langle z, \mathsf{dist}_b \rangle = \sum_{(i,i')} z_{i,i'} \cdot \mathsf{dist}_b(i,i') = \sum_{(i,i'):\, b(i) \neq b(i')} z_{i,i'}. \tag{3}$$

Thus, the left hand side of Equation (2) is the maximum cut in G_z, as in Conjecture 3.

The RHS of Equation (2). We view the code used by Alice in her first message as a sequence of one-bit functions. Since in our protocol this code is randomly sampled, each of the 2^m one-bit functions is expected to appear equally often in Alice's message[9]. As the channel can corrupt each of these one-bit functions independently of all the others, we get that a distance function dist can be induced by the corruptions inserted in Alice's first message (i.e., $\mathsf{dist} \in Q$) if and only if it is the expectation (under the uniform distribution over one-bit functions) of the distance functions that can be induced by corrupting one-bit functions.

Now, if Alice is sending a one-bit function $b : [m] \to \{0,1\}$, there are only two possibilities for Bob: either he receives a 0 or he receives a 1. Let dc_b be the distance contribution function dictated by Bob's received bit. We next show that in the former case, where Bob receives 0, the value of $\langle z, \mathsf{dist}_{\mathsf{dc}_b} \rangle$ is the value of the cut in G_z indicated by b plus twice the weight of all edges such that $b(\cdot) = 0$ on both its endpoints. To see that, recall that $\mathsf{dist}_{\mathsf{dc}_b}(i,i') = \mathsf{dc}_b(i) + \mathsf{dc}_b(i')$ and that we assume $\mathsf{dc}_b(y) = 1 - \mathsf{N}(y)$. In our case, $\mathsf{dc}_b(i) = 1 - 0 = 1$ if $b(i) = 0$ (Alice's bit was not corrupted) and $\mathsf{dc}_b(i) = 0$ if $b(i) = 1$ (Alice's bit was corrupted). This implies that $\mathsf{dist}_{\mathsf{dc}_b}(i,i') = 0$ if $b(i) = b(i') = 1$, and that $\mathsf{dist}_{\mathsf{dc}_b}(i,i') = 1$ if $b(i) \neq b(i')$, and that $\mathsf{dist}_{\mathsf{dc}_b}(i,i') = 2$ if $b(i) = b(i') = 0$. Therefore,

$$\max_{\mathsf{dist} \in Q} \langle z, \mathsf{dist}_{\mathsf{dc}_b} \rangle = \sum_{(i,i')} z_{i,i'} \cdot \mathsf{dist}_{\mathsf{dc}_b}(i,i') = \sum_{(i,i'):\, b(i) \neq b(i')} z_{i,i'} + \sum_{(i,i'):\, b(i) = b(i') = 0} 2 \cdot z_{i,i'}. \tag{4}$$

Similarly, it can be shown that in the latter case, where Bob gets 1, the value of $\langle z, \mathsf{dist}_{\mathsf{dc}_b} \rangle$ is the value of the cut in G_z indicated by b plus twice the weight of all edges such that $b(\cdot) = 1$ on both its endpoints.

[9] We mention that this is only in expectation and the length of Alice's message need not depend exponentially on m.

Recall that the bit function b is a uniformly random bit-function. Taking an expectation over one bit functions b, the value of the cut in G_z indicated by b is exactly the constant $\frac{1}{2}$ and the other terms on the right hand side of Equation (3) and Equation (4) are exactly as on the right hand side of Conjecture 3, where the maximum becomes minimum because of the constants involved. Equation (2) now directly follows from Conjecture 3.

2.2.2 A Tight Protocol Implies Conjecture 3

We now finish this sketch by arguing why a tight protocol implies Conjecture 3. For this, we note that all the arguments in Section 2.2.1 were actually equivalences, except two, one of which was explicitly stated and one was not. The explicit one was our assumption that Alice's first message is simply the encoding of her input using a randomly sampled code. The second one was that Alice gets feedback from Bob at round $\frac{8T}{23}$, where T is the total number of rounds of the protocol. The constant $\frac{8}{23}$ may seem arbitrary, but it is the constant one gets when one tries to match the constants obtained in the analysis in Section 2.2.1 with the constants in Conjecture 3.

Both these assumptions are actually without loss of generality. We start by arguing this for the second one, again ignoring the actual constants and only stating the high level idea. Roughly speaking, the second assumption is without loss of generality as Conjecture 3 is tight for cliques of size 3 and 5, and if the constant is anything other than $\frac{8}{23}$, Equation (2) will fail to hold for z corresponding to one of these cliques.

It remains to show why the first assumption is without loss of generality. For this, our approach is to take an arbitrary code $C : [m] \to \{0,1\}^*$ that Alice may use for her first message, and in several steps, convert it to a code that looks more and more like a random code, at the cost of a smaller m. In each step k, we convert C to a code that is k-random, in the sense that any set of k codewords of the new code looks like k codewords from a randomly sampled code.

For $k = 1$, this means that we have to show that each codeword has an equal number of 0s and 1s, and this can be easily achieved by concatenating all codewords with their negations (which preserves the distance properties). We now show how to get a 2-random code from a 1-random code, noting that similar (but technically more involved) ideas allow us to get a $(k+1)$-random code from a k-random code, for any $k \geq 1$. To show that a code is 2-random, we need to show that it is 1-random and that the fractional distance between any pair of codewords is (roughly) $\frac{1}{2}$.

For this, let $\epsilon > 0$ be an error parameter and construct a complete graph with the m codewords as the vertices, and color the edge between codewords i and i' as (1) red, if the fractional distance between them is smaller than $\frac{1}{2} - \epsilon$, (2) blue, if the fractional distance between them is between $\frac{1}{2} - \epsilon$ and $\frac{1}{2} + \epsilon$, (3) green, if the fractional distance between them is larger than $\frac{1}{2} + \epsilon$. As m gets larger and larger, Ramsey theory tells us that there must exist a large (going to infinity with m) monochromatic clique in this graph. This clique cannot be red, as we show that a large number of pairwise close codewords can be used to break the protocol. It also cannot be green, as that would violate known distance bounds for error correcting codes. Thus, it must be blue, implying that restricting attention to this clique gives us our desired 2-random code.

3 Model and Preliminaries

3.1 Notation and Preliminaries

For $x \in \mathbb{R}$, let $\overrightarrow{x}$ be the vector (of appropriate dimension inferred from context) with all its coordinates being x. Throughtout, all inequalities between vectors are coordinate-wise. For $k \geq 0$, $\Delta^k = \{(x_0, \ldots, x_k) \in \mathbb{R}^{k+1} \mid \sum_{i=0}^{k} x_i = 1 \text{ and } x_i \geq 0 \text{ for all } i \in [0,k]\}$ denotes the k-dimensional standard simplex. For $x \in \mathbb{R}$ and $k \geq 0$, we write $x^{\underline{k}}$ as a shorthand for falling factorial $\prod_{i=0}^{k-1}(x-i)$.[10] For a set S and $k \geq 0$, let $\binom{S}{k}$ be the collection of all subsets of S of size k. For a function $f : X \to Y$ and subset $X' \subseteq X$, $f|_{X'}$ denotes the restriction of f onto X'. For $x, y \geq 1$, $\mathsf{R}(x,y)$ is the (two-color) Ramsey number for x, y, which is well-known to be finite. For $k \geq 1$ and two bit strings $x, y \in \{0,1\}^k$, their Hamming distance is $\Delta(x,y) = \sum_{i=1}^{k} \mathbb{1}[x_i \neq y_i]$.

3.2 Our Model: Round-Restricted Binary Feedback Channels

We now define (deterministic, binary) *protocols* with (non-adaptive) round-restricted feedback for the message transfer task, where Alice has an input and Bob's goal is to learn this input. Such a protocol is defined by a tuple:

$$\Pi = \Big(n, r, \{L_i\}_{i\in[r+1]}, \{f_i\}_{i\in[r+1]}, \mathsf{out}\Big), \tag{5}$$

where (1) $\{0,1\}^n$ is the set of all possible inputs for Alice; (2) r is the number of feedback rounds. Equivalently, we can say that Alice speaks in $r+1$ rounds; (3) For all $i \in [r+1]$, L_i is the length of Alice's message in the i-th round. Throughout, we use $L = \sum_{i=1}^{r+1} L_i$; (4) For all $i \in [r+1]$, $f_i : \{0,1\}^n \times \{0,1,\perp\}^{L_1} \times \cdots \times \{0,1,\perp\}^{L_{i-1}} \to \{0,1\}^{L_i}$ is the message function Alice uses in the i-th round; (5) $\mathsf{out} : \{0,1,\perp\}^{L_1} \times \cdots \times \{0,1,\perp\}^{L_{r+1}} \to \{0,1\}^n$ is the function Bob uses to compute the output.

Execution of a protocol. Let Π be a protocol as above. An adversary for Π is defined by a function $\mathsf{Adv} : \{0,1\}^n \to \{0,1,\perp\}^{L_1} \times \cdots \times \{0,1,\perp\}^{L_{r+1}}$. For $i \in [r+1]$, we will use $\mathsf{Adv}_i(\cdot)$ to denote the function that outputs the i-th coordinate of $\mathsf{Adv}(\cdot)$. We next define an execution of Π in the presence of an adversary Adv for Π: At the beginning of the execution, Alice starts with an input $x \in \{0,1\}^n$. The execution consists of $r+1$ rounds and before the i-th round, for $i \in [r+1]$, Alice and Bob have the (same) transcript $\tau_{<i} \in \{0,1,\perp\}^{L_1} \times \cdots \times \{0,1,\perp\}^{L_{i-1}}$. In round i, Alice computes the message $f_i(x, \tau_{<i}) \in \{0,1,\perp\}^{L_i}$ and sends it to Bob bit by bit, while Bob receives the string $\tau_i = \mathsf{Adv}_i(x)$. As we assume a feedback channel, if $i \leq r$, Alice also receives the string τ_i and both the parties add τ_i to $\tau_{<i}$ and continue executing the protocol.

If $i = r+1$, the execution of the protocol terminates and Bob outputs $\mathsf{out}(\tau_{\leq r+1})$. Observe that this execution is completely determined by x, Π, and Adv. We denote the output of Π on input x in the presence of adversary Adv by $\mathsf{out}_{\Pi,\mathsf{Adv}}(x)$.

Counting the noise. Let Π be a protocol as above and Adv be an adversary for Π. For $x \in \{0,1\}^n$, the amount of noise added by Adv in Π on input x is the number of times Bobs' received bit is different from the bit Alice sent. Formally, we have:

$$\mathsf{noise}_{\Pi,\mathsf{Adv}}(x) = \sum_{i=1}^{r+1} \Delta(\mathsf{Adv}_i(x), f_i(x, \mathsf{Adv}_{<i}(x))). \tag{6}$$

[10] See also Falling factorials (Wikipedia) for the notation.

For $\theta \in [0,1]$, we say that an adversary Adv has budget θ if we have

$$\max_{x \in \{0,1\}^n} \mathsf{noise}_{\Pi,\mathsf{Adv}}(x) \leq \theta L.$$

Types of Adversaries. Let Π be a protocol as above and Adv be an adversary for Π. We say that Adv is a *corruption adversary* if it never outputs the symbol $\perp$, i.e., for all $x \in \{0,1\}^n$ and all $i \in [r+1]$, we have $\mathsf{Adv}_i(x) \in \{0,1\}^{L_i}$. We say that Adv is an *erasure adversary* if it only "erases" the symbols sent by Alice. More precisely, we say that Adv is an erasure adversary if for all $x \in \{0,1\}^n$, all $i \in [r+1]$, and all $j \in [L_i]$, if $(\mathsf{Adv}_i(x))_j \neq \perp$, then we have $(\mathsf{Adv}_i(x))_j = (f_i(x, \mathsf{Adv}_{<i}(x)))_j$.

Resilience of a protocol. Let Π be a protocol as above and $\theta \in [0,1]$. We say that Π *has resilience* θ over the binary erasure channel if for all erasure adversaries with budget θ and all $x \in \{0,1\}^n$, it holds that $\mathsf{out}_{\Pi,\mathsf{Adv}}(x) = x$. Resilience over the binary corruption channel is defined analogously.

4 Optimal List-Decodable Small Codes

In this section, we construct the codes used by our protocol.

4.1 Definitions of List Decodability

Codes for erasures. We start by defining list decodability for erasures.

▶ **Definition 6.** *Let $m, k, L \geq 1$ and $d \in [0,1]$. We say that a code $C : [m] \to \{0,1\}^L$ is less-than-k-list decodable for erasures up to radius d if for all subsets $\Gamma \in \binom{[m]}{k}$, we have $\mathsf{ns}_C(\Gamma) > d$, where:*

$$\mathsf{ns}_C(\Gamma) = 1 - \frac{1}{L} \cdot \sum_{j=1}^{L} \mathbb{1}[\exists b \in \{0,1\} \ \forall i \in \Gamma : C_j(i) = b].$$

To get the intuition behind the definition of ns, observe that $\mathsf{ns}_C(\Gamma)$ is the minimum fraction e of erasures for which there exists $\tau \in \{0,1,\perp\}^L$ such that for all $i \in \Gamma$, it is possible to erase $e \cdot L$ symbols from $C(i)$ and get τ. Observe that this is equal to the fraction of coordinates where the encodings $\{C(i)\}_{i \in \Gamma}$ are *not* all the same (ns = not same).

For $m, k \geq 1$, we define $\mathsf{d}_{\mathsf{erase}}(m,k)$ to be the supremum of all values $d \in [0,1]$ for which there exists $L \geq 1$ and a code $C : [m] \to \{0,1\}^L$ that is less-than-k-list decodable for erasures up to radius d.

Codes for corruptions. Next, we define list decodability for corruptions:

▶ **Definition 7.** *Let $m, k, L \geq 1$ and $d \in [0,1]$. We say that a code $C : [m] \to \{0,1\}^L$ is less-than-k-list decodable for corruptions up to radius d if for all $\tilde{x} \in \{0,1\}^L$, we have*

$$|\{i \in [m] : \Delta(C(i), \tilde{x}) < dL\}| < k.$$

Analogous to $\mathsf{d}_{\mathsf{erase}}$, for $m, k \geq 1$, we define $\mathsf{d}_{\mathsf{corr}}(m,k)$ to be the supremum of all values $d \in [0,1]$ for which there exists $L \geq 1$ and a code $C : [m] \to \{0,1\}^L$ that is less-than-k-list decodable for corruptions up to radius d.

4.2 Lemmas about $\mathsf{d}_{\mathsf{erase}}$ and $\mathsf{d}_{\mathsf{corr}}$

In this section, we show the results we need about $\mathsf{d}_{\mathsf{erase}}$ and $\mathsf{d}_{\mathsf{corr}}$. First, we define a helper function $\mathsf{d}(\cdot,\cdot)$:

$$\mathsf{d}(m,k) = 1 - \frac{\binom{\lfloor m/2 \rfloor}{k} + \binom{\lceil m/2 \rceil}{k}}{\binom{m}{k}}. \tag{7}$$

We now show some useful properties of the function d defined in Equation (7).

▷ Claim 8. The following hold:

1. For all $m \geq k \geq 1$, it holds that
$$\mathsf{d}(m,k) = 1 - \frac{(\lceil m/2 \rceil - 1)^{\underline{k-1}}}{(2\lceil m/2 \rceil - 1)^{\underline{k-1}}}.$$
2. For all $m_2 \geq m_1 \geq k \geq 1$, it holds that $\mathsf{d}(m_2,k) \leq \mathsf{d}(m_1,k)$, and moreover,
$$\lim_{m\to\infty} \mathsf{d}(m,k) = 1 - \frac{1}{2^{k-1}}.$$
It follows that $\mathsf{d}(m,k) \geq 1 - \frac{1}{2^{k-1}}$ for all $m \geq k \geq 1$.
3. For all $m \geq k_2 \geq k_1 \geq 1$, it holds that $\mathsf{d}(m,k_2) \geq \mathsf{d}(m,k_1)$.

4.2.1 Lemmas about $\mathsf{d}_{\mathsf{erase}}$

We now show that the functions $\mathsf{d}_{\mathsf{erase}}$ and d are the exact same. Owing to this lemma, we omit writing erase in the subscript in the rest of this text.

▶ **Lemma 9.** *For all $m,k \geq 1$, we have:*

$$\mathsf{d}_{\mathsf{erase}}(m,k) = \mathsf{d}(m,k).$$

▶ **Lemma 10.** *For all $\epsilon > 0$, there exists a constant K such that for all $K' \geq K$ and for all $m \geq 1$, there exists a code $C : [m] \to \{0,1\}^{K' \log m}$ such that for all $k \in [m]$, the code C is less-than-k-list decodable up to radius $\mathsf{d}(m,k) - \epsilon$.*

4.2.2 Lemmas about $\mathsf{d}_{\mathsf{corr}}$

Using the results of Section 4.2.1, we show the following lemma:

▶ **Lemma 11.** *For all $m,k \geq 2$, we have:*

$$\mathsf{d}_{\mathsf{corr}}(m,2) = \frac{\mathsf{d}(m,2)}{2} \qquad \text{and} \qquad \lim_{m\to\infty} \mathsf{d}_{\mathsf{corr}}(m,k) = \frac{1}{2} - \frac{\binom{k-1}{\lceil k/2 \rceil - 1}}{2^k}.$$

5 Protocols Against Erasures

In this section, we show one direction of Theorem 1, as formalized below. Later, in Section 6, we prove the other direction.

▶ **Theorem 12.** *For all $\epsilon > 0$ and $r,n \in \mathbb{N}$, there exists a constant-rate (polynomial in ϵ) protocol for message transfer with r rounds of feedback, input length n, and the following resilience over the binary erasure channel:*

$$\begin{cases} \frac{5}{7} - \epsilon, & \text{if } r = 1 \\ 1 - \frac{7}{12(r+1)} - \epsilon, & \text{if } r > 1 \end{cases}.$$

We prove Theorem 12 in the rest of this section. Throughout, we fix $\epsilon > 0$ and $r, n \in \mathbb{N}$. We assume $r < \frac{10}{\epsilon}$. This is without loss of generality as a protocol for large r follows from a protocol for smaller r.

5.1 Our Protocol

Let K be the constant from Lemma 10 for ϵ. For all $K' \geq K$ and all $m \geq 1$, let $C_{m,K'} : [m] \to \{0,1\}^{K' \log m}$ be as promised by Lemma 10. We will omit K' when it is clear from context. For a set Γ of size m, we will also view C_m as a code $C_\Gamma : \Gamma \to \{0,1\}^{K' \log m}$. Our protocol is given in Algorithm 1, where the lengths of the rounds are given as follows:

$$L_i = \begin{cases} \frac{4}{3} \cdot Kn, & \text{if } i = r = 1 \\ Kn, & \text{otherwise} \end{cases}. \tag{8}$$

Algorithm 1 Message transfer protocol over the erasure channel with $r \geq 1$ feedback rounds.

Input: Alice has input $x \in \Gamma_0 = \{0,1\}^n$.
Output: Bob outputs $y \in \{0,1\}^n$.

1: **for** $i = 1, \ldots, r+1$ **do**
2: Alice sends $C_{\Gamma_{i-1}}(x) \in \{0,1\}^{L_i}$ bit by bit.
3: Bob receives $\tau_i \in \{0,1,\perp\}^{L_i}$ and sends τ_i via the noiseless feedback channel.
4: Bob computes

$$\Gamma_i = \{x' \in \Gamma_{i-1} \mid \forall j \in [L_i] : \tau_{i,j} \in \{C_{\Gamma_{i-1},j}(x'), \perp\}\}.$$

5: If $i \leq r$, Alice receives τ_i as feedback and also computes Γ_i as above.
6: **end for**
7: Bob outputs the lexicographically first element in Γ_{r+1}, aborting if $\Gamma_{r+1} = \emptyset$.

5.2 Analysis

We now analyze Algorithm 1 and finish proving Theorem 12. That the protocol is constant rate is clear from Algorithm 1. It remains to show that it has the claimed noise resilience. For this, we fix an input x for Alice and an erasure adversary Adv for the protocol with desired budget as in Theorem 12. Observe that fixing x and Adv fixes the value of all the variables in the execution of Algorithm 1. For the analysis, we first show that:

▶ **Lemma 13.** *For all $i \in [0, r+1]$, we have $x \in \Gamma_i$.*

▶ **Lemma 14.** *For all $m \geq k' \geq k \geq 2$ such that $(k', k) \neq (3, 2)$, it holds that*

$$\mathsf{d}(m, k') + \mathsf{d}(k', k) \geq 1 + \mathsf{d}(m, k).$$

At a high level, Lemma 14 will be applied as follows: Consider an adversary that shrinks the set Γ from size m to size k' in a given round and from size k' to size k in the next round. Lemma 14 shows that, if the two rounds are of equal length (recall from Equation (8) that the round lengths are always the same except when $i = r = 1$), then it is always better for the adversary to erase one of the rounds completely and shrink from size m to size k directly in the other round.

We now divide the proof into two cases based on whether or not $r = 1$.

5.2.1 Proof of Theorem 12 When $r = 1$

Let $k_i = |\Gamma_i|$ for $i \in [0, 2]$. We prove the theorem by showing that $k_2 \leq 1$. Together with Lemma 13 and Algorithm 1, this shows the correctness of Algorithm 1.

Observe that at the beginning of the i-th round, Alice and Bob agree on Γ_{i-1}, the subset of all remaining possibilities for x from the perspective of Bob, that are still consistent with the partial transcript $\tau_1, \ldots, \tau_{i-1}$ so far. In order to keep Bob confused among Γ_i, the adversary has to erase at least a $\mathsf{d}(k_{i-1}, k_i) - \epsilon$ fraction of Alice's i-th message, due to Lemma 10. As this holds for all rounds, the overall fraction of erasures is lower bounded by

$$\frac{4}{7}(\mathsf{d}(k_0, k_1) - \epsilon) + \frac{3}{7}(\mathsf{d}(k_1, k_2) - \epsilon) = \frac{4\mathsf{d}(k_0, k_1) + 3\mathsf{d}(k_1, k_2)}{7} - \epsilon.$$

Now suppose $k_2 \geq 2$. It is sufficient to show $4\mathsf{d}(k_0, k_1) + 3\mathsf{d}(k_1, k_2) \geq 5$ for a contradiction. Without loss of generality, we assume $k_2 = 2$ since $\mathsf{d}(k_1, k_2)$ only decrease as k_2 becomes smaller by Item 3 of Claim 8. If $k_1 = 3$, we have

$$4\mathsf{d}(k_0, k_1) + 3\mathsf{d}(k_1, k_2) = 4\mathsf{d}(k_0, 3) + 3\mathsf{d}(3, 2) \geq 4 \cdot \frac{3}{4} + 3 \cdot \frac{2}{3} = 5$$

by Item 2 of Claim 8. Otherwise, by Lemma 14, we also get

$$\begin{aligned}
4\mathsf{d}(k_0, k_1) + 3\mathsf{d}(k_1, k_2) &= 4(\mathsf{d}(k_0, k_1) + \mathsf{d}(k_1, 2)) - \mathsf{d}(k_1, 2) \\
&\geq 4(1 + \mathsf{d}(k_0, 2)) - \mathsf{d}(k_1, 2) \\
&\geq 3 + 4\mathsf{d}(k_0, 2) && \text{(as } \mathsf{d}(\cdot, \cdot) \text{ is always upper bounded by 1)} \\
&\geq 3 + 4 \cdot \frac{1}{2} && \text{(by Item 2 of Claim 8)} \\
&= 5.
\end{aligned}$$

5.2.2 Proof of Theorem 12 When $r > 1$

Let $k_i = |\Gamma_i|$ for $i \in [0, r+1]$. Similarly to the proof in Section 5.2.1, we show that $k_{r+1} \leq 1$. This ensures Bob outputs the correct $y = x$ because of Lemma 13 and Algorithm 1. Using a similar argument to Section 5.2.1, we have that the overall fraction of erasures is lower bounded by

$$\frac{1}{r+1} \cdot \sum_{i=1}^{r+1} \mathsf{d}(k_{i-1}, k_i) - \epsilon.$$

Now for the purpose of contradiction, suppose that $k_{r+1} \geq 2$. It is sufficient to show

$$\frac{1}{r+1} \cdot \sum_{i=1}^{r+1} \mathsf{d}(k_{i-1}, k_i) \geq 1 - \frac{7}{12(r+1)}.$$

In the following, we again assume without loss of generality that $k_{r+1} = 2$ since $\mathsf{d}(k_r, k_{r+1})$ decreases as k_{r+1} becomes smaller by Item 3 of Claim 8. Let $j = \min\{t \in [r+1] \mid k_t \leq 3\}$. By repeatedly applying Lemma 14, we have

$$\begin{aligned}
&\frac{1}{r+1}\cdot\sum_{i=1}^{r+1}\mathsf{d}(k_{i-1},k_i)\\
&\qquad\geq\frac{1}{r+1}\cdot\left(1+\mathsf{d}(k_0,k_2)+\sum_{i=3}^{r+1}\mathsf{d}(k_{i-1},k_i)\right)\\
&\qquad\quad\vdots\\
&\qquad\geq\frac{1}{r+1}\cdot\left(j-1+\mathsf{d}(k_0,k_j)+\sum_{i=j+1}^{r+1}\mathsf{d}(k_{i-1},k_i)\right).
\end{aligned}$$

Since $k_0\geq\cdots\geq k_{r+1}=2$ by definition of Γ_i, either $k_j=2$ or $k_j=3$.

In the former case where $k_j=2$, we also have $k_{j+1}=\cdots=k_{r+1}=2$ and thus

$$\begin{aligned}
&\frac{1}{r+1}\cdot\sum_{i=1}^{r+1}\mathsf{d}(k_{i-1},k_i)\\
&\qquad\geq\frac{1}{r+1}\cdot\left(j-1+\mathsf{d}(k_0,k_j)+\sum_{i=j+1}^{r+1}\mathsf{d}(k_{i-1},k_i)\right)\\
&\qquad=\frac{1}{r+1}\cdot\left(j-1+\mathsf{d}(k_0,2)+(r+1-j)\cdot\mathsf{d}(2,2)\right)\\
&\qquad\geq\frac{1}{r+1}\cdot\left(j-1+\frac{1}{2}+r+1-j\right)\qquad\text{(by Item 2 of Claim 8)}\\
&\qquad=1-\frac{1}{2(r+1)}\\
&\qquad\geq1-\frac{7}{12(r+1)}.
\end{aligned}$$

In the latter case where $k_j=3$, let $j'=\min\{t\in[r+1]\mid k_t=2\}$. Then we have

$$\begin{aligned}
&\frac{1}{r+1}\cdot\sum_{i=1}^{r+1}\mathsf{d}(k_{i-1},k_i)\\
&\qquad\geq\frac{1}{r+1}\cdot\left(j-1+\mathsf{d}(k_0,k_j)+\sum_{i=j+1}^{r+1}\mathsf{d}(k_{i-1},k_i)\right)\\
&\qquad=\frac{1}{r+1}\cdot\left(j-1+\mathsf{d}(k_0,3)+(j'-1-j)\cdot\mathsf{d}(3,3)+\mathsf{d}(3,2)+(r+1-j')\cdot\mathsf{d}(2,2)\right)\\
&\qquad\geq\frac{1}{r+1}\cdot\left(j-1+\frac{3}{4}+j'-1-j+\frac{2}{3}+r+1-j'\right)\qquad\text{(by Item 2 of Claim 8)}\\
&\qquad=1-\frac{7}{12(r+1)}.
\end{aligned}$$

This concludes the proof.

6 Impossibility Result for Erasures

In this section, we show the other direction of Theorem 1, as formalized below.

▶ **Theorem 15.** *For all $r \in \mathbb{N}$, there exists an $n \in \mathbb{N}$ such that the resilience of any protocol for message transfer with r rounds of feedback and input length n over the binary erasure channel is at most:*

$$\begin{cases} \frac{5}{7}, & \text{if } r = 1 \\ 1 - \frac{7}{12(r+1)}, & \text{if } r > 1 \end{cases}.$$

We prove Theorem 15 in the rest of this section. Throughout, we work with a fixed $r \in \mathbb{N}$ and define n to be large enough for asymptotic inequalities to hold. We now divide the proof into two cases based on whether or not $r = 1$.

6.1 Proof of Theorem 15 When $r = 1$

Fix a protocol Π with input length n and one round of feedback. Recall Equation (5) and let L_1, L_2 be the lengths of Alice's messages sent in the two rounds, and $f_1 : \{0,1\}^n \to \{0,1\}^{L_1}$, $f_2 : \{0,1\}^n \times \{0,1,\perp\}^{L_1} \to \{0,1\}^{L_2}$ be the two message functions Alice uses in the two rounds.

First suppose that $L_1 \geq \frac{4}{7}(L_1 + L_2)$. By Lemma 9, there exists a subset $\Gamma = \{x_1, x_2\} \in \binom{\{0,1\}^n}{2}$ such that $\mathsf{ns}_{f_1}(\Gamma) \leq \mathsf{d}(2^n, 2)$. This implies the adversary is able to erase a $\mathsf{d}(2^n, 2)$ fraction of Alice's first message so that Bob's view when Alice's input is x_1 is identical to Bob's view when Alice's input is x_2, and therefore Bob is forced to send the same feedback $\tau_1 \in \{0,1,\perp\}^{L_1}$ in both cases. Now the adversary simply erases Alice's second message entirely implying that Bob can never output the correct answer. By Item 2 of Claim 8, the overall fraction of erasures is upper bounded as

$$\mathsf{d}(2^n, 2) \cdot \frac{L_1}{L_1 + L_2} + 1 \cdot \frac{L_2}{L_1 + L_2} \leq \frac{4 \cdot \mathsf{d}(2^n, 2) + 3}{7} \xrightarrow{n \to \infty} \frac{5}{7}.$$

Now consider the other case where $L_1 \leq \frac{4}{7}(L_1 + L_2)$. Again by Lemma 9, there exists a subset $\Gamma = \{x_1, x_2, x_3\} \in \binom{\{0,1\}^n}{3}$ such that $\mathsf{ns}_{f_1}(\Gamma) \leq \mathsf{d}(2^n, 3)$. In this case, the adversary erases a $\mathsf{d}(2^n, 3)$ fraction of Alice's first message so that Bob's view is the same when Alice's input is any of x_1, x_2, x_3. Bob must send the same feedback $\tau_1 \in \{0,1\}^{L_1}$ in all three cases. Note that $f_2(\cdot, \tau_1)$ can also be viewed as a valid code and thus Lemma 9 still applies. In particular, it is always possible to erase a $\mathsf{d}(3, 2) = \frac{2}{3}$ fraction of Alice's second message so that for at least two of x_1, x_2, x_3, Bob's view remains the same at the end of the protocol. This concludes the proof as the overall fraction of erasures is at most

$$\mathsf{d}(2^n, 3) \cdot \frac{L_1}{L_1 + L_2} + \frac{2}{3} \cdot \frac{L_2}{L_1 + L_2} \leq \frac{4 \cdot \mathsf{d}(2^n, 3) + 3 \cdot \frac{2}{3}}{7} \xrightarrow{n \to \infty} \frac{5}{7}.$$

6.2 Proof of Theorem 15 When $r > 1$

Fix a protocol Π with input length n and r rounds of feedback. Recall Equation (5) and for $t \in [r+1]$, let L_t be the length of Alice's message sent in the t-th round. Let $L = \sum_{t=1}^{r+1} L_t$.

We prove the theorem using an approach similar to Section 6.2, i.e., the adversary is always able to erase Alice's messages in such a way that Bob has the same view at the end of the protocol for at least two different inputs. In particular, the adversary erases the entire messages of all rounds except for $i = \arg\max_{t \in [r+1]} L_t$, the longest round, and $j = \arg\max_{t \in [r+1] \setminus \{i\}} L_t$, the second longest round. Then we have

$$L_i \geq \frac{L}{r+1}, \tag{9}$$

$$L_j \geq \frac{L - L_i}{r}. \tag{10}$$

First consider the case where $i < j$. Since the first $i-1$ rounds are completely erased, Bob obviously has the same view for all possible inputs at the beginning of the i-th round. By Lemma 9, the adversary can erase a $\mathsf{d}(2^n, 3)$ fraction of Alice's i-th message so that Bob's view is the same when Alice's input is any of some subset $\Gamma = \{x_1, x_2, x_3\} \subseteq \{0,1\}^n$. This remains true at the beginning of the j-th round as all intermediate rounds are completely erased. Now again by Lemma 9, the adversary is able to erase a $\mathsf{d}(3,2) = \frac{2}{3}$ fraction of Alice's j-th message so that Bob still has the same view at the end of the j-th round, for at least two of x_1, x_2, x_3. As all remaining rounds are also completely erased, Bob can never output the correct answer at the end of the protocol. By Item 2 of Claim 8, the overall fraction of erasures is upper bounded as

$$\begin{aligned}
&\mathsf{d}(2^n,3)\cdot\frac{L_i}{L} + \frac{2}{3}\cdot\frac{L_j}{L} + 1\cdot\frac{L - L_i - L_j}{L} \\
&\quad= 1 - \big(1 - \mathsf{d}(2^n,3)\big)\cdot\frac{L_i}{L} - \frac{1}{3}\cdot\frac{L_j}{L} \\
&\quad\leq 1 - \big(1 - \mathsf{d}(2^n,3)\big)\cdot\frac{L_i}{L} - \frac{1}{3r}\cdot\frac{L - L_i}{L} && \text{(by Equation (10))} \\
&\quad= 1 - \frac{1}{3r} - \left(1 - \mathsf{d}(2^n,3) - \frac{1}{3r}\right)\cdot\frac{L_i}{L} \\
&\quad\leq 1 - \frac{1}{3r} - \left(1 - \mathsf{d}(2^n,3) - \frac{1}{3r}\right)\cdot\frac{1}{r+1} \\
&\qquad\qquad \text{(as } 1 - \mathsf{d}(2^n,3) \xrightarrow{n\to\infty} \tfrac{1}{4} > \tfrac{1}{3r} \text{ for } r \geq 2\text{, and by Equation (9))} \\
&\quad\xrightarrow{n\to\infty} 1 - \frac{7}{12(r+1)}.
\end{aligned}$$

Now suppose that $i > j$. In this case, a similar argument shows the adversary must be able to confuse Bob by erasing a $\mathsf{d}(2^n,3)$ fraction of Alice's j-th message as well as a $\mathsf{d}(3,2) = \frac{2}{3}$ fraction of Alice's i-th message (in addition to completely erasing all other rounds of messages). Observe that $L_i \geq L_j$ by definition and that $\mathsf{d}(2^n,3) \xrightarrow{n\to\infty} \frac{3}{4} \geq \frac{2}{3}$. So the overall fraction of erasures is at most

$$\mathsf{d}(2^n,3)\cdot\frac{L_j}{L} + \frac{2}{3}\cdot\frac{L_i}{L} + 1\cdot\frac{L - L_i - L_j}{L} \leq \mathsf{d}(2^n,3)\cdot\frac{L_i}{L} + \frac{2}{3}\cdot\frac{L_j}{L} + \frac{L - L_i - L_j}{L},$$

which has the desired upper bound as already shown above.

References

1 Rudolf Ahlswede, Christian Deppe, and Vladimir S. Lebedev. Non-binary error correcting codes with noiseless feedback, localized errors, or both. In *International Symposium on Information Theory (ISIT)*, pages 2486–2487, 2006. doi:10.1109/ISIT.2006.262057.

2 Noga Alon. Voting paradoxes and digraphs realizations. *Adv. Appl. Math.*, 29(1):126–135, 2002. doi:10.1016/S0196-8858(02)00007-6.

3 Noga Alon, Boris Bukh, and Yury Polyanskiy. List-decodable zero-rate codes. *IEEE Trans. Inf. Theory*, 65(3):1657–1667, 2019. doi:10.1109/TIT.2018.2868957.

4 Elwyn R. Berlekamp. *Block coding with noiseless feedback*. PhD thesis, Massachusetts Institute of Technology, USA, 1964. URL: http://hdl.handle.net/1721.1/14783.

5 Elwyn R Berlekamp. Block coding for the binary symmetric channel with noiseless, delayless feedback. *Error-correcting codes*, pages 61–68, 1968.

6 Vladimir Markovich Blinovskii. Bounds for codes in the case of finite-volume list decoding. *Problemy Peredachi Informatsii*, 22(1):11–25, 1986.

7 Vladimir M Blinovsky. Plotkin bound generalization to the case of multiple packings. *Problems of Information Transmission*, 45(1):1–4, 2009. doi:10.1134/S0032946009010013.

8 Mark Braverman, Klim Efremenko, Gillat Kol, Raghuvansh Saxena, and Zhijun Zhang. Round-vs-resilience tradeoffs for binary feedback channels. *Electron. Colloquium Comput. Complex.*, TR22-179, 2022. URL: https://eccc.weizmann.ac.il/report/2022/179, arXiv:TR22-179.

9 Marat Valievich Burnashev. Data transmission over a discrete channel with feedback. random transmission time. *Problemy peredachi informatsii*, 12(4):10–30, 1976.

10 Klim Efremenko, Gillat Kol, and Raghuvansh R. Saxena. Binary interactive error resilience beyond 1/8 (or why $(1/2)^3 > 1/8$). In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 470–481. IEEE, 2020. doi:10.1109/FOCS46700.2020.00051.

11 Klim Efremenko, Gillat Kol, Raghuvansh R. Saxena, and Zhijun Zhang. Binary codes with resilience beyond 1/4 via interaction. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 1–12. IEEE, 2022. doi:10.1109/FOCS54457.2022.00008.

12 Peter Elias. List decoding for noisy channels. Technical report, Research Laboratory of Electronics, Massachusetts Institute of Technology, 1957.

13 Krishnan Eswaran, Anand D. Sarwate, Anant Sahai, and Michael Gastpar. Zero-rate feedback can achieve the empirical capacity. *IEEE Transactions on Information Theory*, 56(1):25–39, 2010. doi:10.1109/TIT.2009.2034779.

14 David G. Forney. Exponential error bounds for erasure, list, and decision feedback schemes. *IEEE Transactions on Information Theory*, 14(2):206–220, 1968. doi:10.1109/TIT.1968.1054129.

15 Meghal Gupta, Venkatesan Guruswami, and Rachel Yun Zhang. Binary error-correcting codes with minimal noiseless feedback. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1475–1487. ACM, 2023. doi:10.1145/3564246.3585126.

16 Meghal Gupta, Yael Tauman Kalai, and Rachel Yun Zhang. Interactive error correcting codes over binary erasure channels resilient to $>$ ½ adversarial corruption. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 609–622. ACM, 2022. doi:10.1145/3519935.3519980.

17 Meghal Gupta and Rachel Yun Zhang. Efficient interactive coding achieving optimal error resilience over the binary channel. *CoRR*, abs/2207.01144, 2022. doi:10.48550/arXiv.2207.01144.

18 Meghal Gupta and Rachel Yun Zhang. The optimal error resilience of interactive communication over binary channels. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 948–961. ACM, 2022. doi:10.1145/3519935.3519985.

19 Meghal Gupta and Rachel Yun Zhang. Positive rate binary interactive error correcting codes resilient to $> 1/2$ adversarial erasures. *CoRR*, abs/2201.11929, 2022. arXiv:2201.11929, doi:10.48550/arXiv.2201.11929.

20 Venkatesan Guruswami. List decoding from erasures: bounds and code constructions. *IEEE Trans. Inf. Theory*, 49(11):2826–2833, 2003. doi:10.1109/TIT.2003.815776.

21 Venkatesan Guruswami and Madhu Sudan. List decoding algorithms for certain concatenated codes. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 181–190, 2000. doi:10.1145/335305.335327.

22 Gregory Z. Gutin and Anders Yeo. Lower bounds for maximum weighted cut. *CoRR*, abs/2104.05536, 2021. doi:10.48550/arXiv.2104.05536.

23 Bernhard Haeupler, Pritish Kamath, and Ameya Velingker. Communication with partial noiseless feedback. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, volume 40 of *LIPIcs*, pages 881–897, 2015. doi:10.4230/LIPIcs.APPROX-RANDOM.2015.881.

24 Michael Horstein. Sequential transmission using noiseless feedback. *IEEE Transactions on Information Theory*, 9(3):136–143, 1963. doi:10.1109/TIT.1963.1057832.

25 M. Plotkin. Binary codes with specified minimum distance. *IRE Transactions on Information Theory*, 6(4):445–450, 1960. doi:10.1109/TIT.1960.1057584.

26 Svatopluk Poljak and Daniel Turzík. A polynomial time heuristic for certain subgraph optimization problems with guaranteed worst case bound. *Discrete Mathematics*, 58(1):99–104, 1986. doi:10.1016/0012-365X(86)90192-5.

27 Anant Sahai. Why do block length and delay behave differently if feedback is present? *IEEE Transactions on Information Theory*, 54(5):1860–1886, 2008. doi:10.1109/TIT.2008.920339.

28 Leonard J Schulman. Communication on noisy channels: A coding theorem for computation. In *Foundations of Computer Science (FOCS)*, pages 724–733, 1992. doi:10.1109/SFCS.1992.267778.

29 Leonard J Schulman. Deterministic coding for interactive communication. In *Symposium on Theory of computing (STOC)*, pages 747–756, 1993. doi:10.1145/167088.167279.

30 Leonard J. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996. doi:10.1109/18.556671.

31 Claude E. Shannon. The zero error capacity of a noisy channel. *IRE Transactions on Information Theory*, 2(3):8–19, 1956. doi:10.1109/TIT.1956.1056798.

32 Ofer Shayevitz. On error correction with feedback under list decoding. In *IEEE International Symposium on Information Theory, ISIT*, pages 1253–1257, 2009. doi:10.1109/ISIT.2009.5205965.

33 Ofer Shayevitz and Meir Feder. Optimal feedback communication via posterior matching. *IEEE Transactions on Information Theory*, 57(3):1186–1222, 2011. doi:10.1109/TIT.2011.2104992.

34 Ofer Shayevitz and Michèle A. Wigger. On the capacity of the discrete memoryless broadcast channel with feedback. *IEEE Transactions on Information Theory*, 59(3):1329–1345, 2013. doi:10.1109/TIT.2012.2227670.

35 Joel Spencer, Peter Winkler, and South St. Three thresholds for a liar. *Combinatorics, Probability and Computing*, 1:81–93, 1992. doi:10.1017/S0963548300000080.

36 Gang Wang, Yanyuan Qin, and Chengjuan Chang. Communication with partial noisy feedback. In *IEEE Symposium on Computers and Communications (ISCC)*, pages 602–607, 2017. doi:10.1109/ISCC.2017.8024594.

37 Norbert Wiener. *Cybernetics or Control and Communication in the Animal and the Machine*. MIT press, 2019.

38 John M Wozencraft. List decoding. *Quarterly Progress Report*, 48:90–95, 1958.

39 K.Sh. Zigangirov. On the number of correctable errors for transmission over a binary symmetrical channel with feedback. *Problems of Information Transmission*, 12:85–97, 1976.

Accumulation Without Homomorphism

Benedikt Bünz
New York University, NY, USA

Pratyush Mishra
University of Pennsylvania, Philadelphia, PA, USA

Wilson Nguyen
Stanford University, CA, USA

William Wang
New York University, NY, USA

Abstract

Accumulation schemes are a simple yet powerful primitive that enable highly efficient constructions of incrementally verifiable computation (IVC). Unfortunately, all prior accumulation schemes rely on homomorphic vector commitments whose security is based on public-key assumptions. It is an interesting open question to construct efficient accumulation schemes that avoid the need for such assumptions.

In this paper, we answer this question affirmatively by constructing an accumulation scheme from *non-homomorphic* vector commitments which can be realized from solely symmetric-key assumptions (e.g., Merkle trees). We overcome the need for homomorphisms by instead performing spot-checks over error-correcting encodings of the committed vectors.

Unlike prior accumulation schemes, our scheme only supports a bounded number of accumulation steps. We show that such *bounded-depth* accumulation still suffices to construct proof-carrying data (a generalization of IVC). We also demonstrate several optimizations to our PCD construction which greatly improve concrete efficiency.

2012 ACM Subject Classification Theory of computation → Cryptographic protocols

Keywords and phrases Proof-carrying data, incrementally verifiable computation, accumulation schemes

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.23

Related Version *Full Version*: `https://eprint.iacr.org/2024/474`

Funding *Benedikt Bünz*: Supported by Alpen Labs, Chaincode Labs, and the Sui Foundation. *Wilson Nguyen*: Supported by NSF, DARPA, the Simons Foundation, UBRI, NTT Research, and the Stanford Future of Digital Currency Initiative (FDCI). Opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

1 Introduction

Proof-carrying data (PCD) [27] is a powerful cryptographic primitive that enables mutually distrustful parties to perform distributed computations that run indefinitely, while ensuring that the correctness of every intermediate step can be verified efficiently. PCD is a generalization of the prior notion of *incrementally verifiable computation* (IVC) [52].[1]

[1] IVC is the special case of PCD where the distributed computation graph is a line.


© Benedikt Bünz, Pratyush Mishra, Wilson Nguyen, and William Wang;
licensed under Creative Commons License CC-BY 4.0



PCD has found numerous applications in both theory and practice, including enforcing language semantics [29], complexity-preserving succinct arguments [10, 9], verifiable MapReduce computations [28], image provenance [42], and consensus protocols and blockchains [44, 38, 14, 24, 19]. It is thus a key question to understand security assumptions that PCD constructions require, as well as the efficiency that they can attain under these assumptions.

Let us review how existing PCD constructions fare along both dimensions. For simplicity, we focus on the special case of IVC in the following discussion.

PCD from succinctly verifiable arguments

The standard construction of PCD is via recursive composition of succinct non-interactive arguments of knowledge (SNARKs) [10, 8, 9, 26]. Informally, to prove a t-step computation, the PCD prover proves that the t-th step is correct, and there exists a valid proof for the first $t-1$ steps. The state-of-the-art works in this line follow the template of Fractal [26] by relying on SNARKs in the random oracle model.[2] This means that we can achieve PCD (heuristically) from only symmetric-key cryptography, but at the cost of relying on the existence of SNARKs, which are complex to construct and incur (asymptotically and concretely) high costs for the PCD prover.

PCD from accumulation

A recent popular approach to avoid this reliance on SNARKs is to construct PCD via *accumulation schemes* [22, 21, 41]. Roughly speaking, instead of recursively checking proofs as above, the PCD prover "accumulates" the proof for each step into a running accumulator, and then the PCD verifier performs a single expensive check on the final accumulator. This line of work has led to simple and efficient PCD schemes that incur low costs for the PCD prover, and so has seen much interest in new constructions and deployments [21, 41, 20, 34, 39, 40]. Unfortunately, all known accumulation schemes rely on homomorphic vector commitments that are only known to exist under public-key assumptions. Additionally, because most existing homomorphic vector commitments are not known to achieve post-quantum security, the resulting accumulation schemes are also quantum-insecure.

Our question

We are thus left in an unsatisfactory state of affairs: on the one hand we have PCD from ROM-based SNARKs that (heuristically) relies on the minimal symmetric-key assumptions, but incurs high PCD prover costs due to this reliance on SNARKs, while on the other hand we have PCD from accumulation that relies on public-key assumptions, but achieves comparatively lower PCD prover costs by avoiding SNARKs. This motivates the questions we tackle in this paper: can we design PCD that achieves low prover costs by avoiding SNARKs, while simultaneously minimizing assumptions?

1.1 Our contributions

We answer this question positively by constructing accumulation schemes *solely* in the random-oracle model. Our constructions satisfy a weaker notion of accumulation than that considered in prior work, but we show that this weaker notion still suffices to construct PCD (again, heuristically after instantiating the random oracle). We provide details below.

[2] The concrete PCD construction makes non-black-box use of the SNARK verifier, which requires us to heuristically instantiate the random oracle.

(1) Bounded-depth accumulation

We introduce a new notion of *bounded-depth* accumulation schemes that can only provide (knowledge) soundness guarantees for a limited number of consecutive accumulation steps. In contrast, prior accumulation schemes support an unbounded number of accumulation steps.

(2) PCD from bounded-depth accumulation

We show that this weaker notion of accumulation still suffices to construct (a variant of) proof-carrying data. In particular, we show that bounded-depth accumulation suffices to construct PCD for computation graphs of a-priori bounded depth $O(1)$, which is already sufficient for many applications, including the primary application of constructing polynomial-length IVC [10].

We note that, like our construction, all prior PCD constructions only provably support computation graphs of bounded depth [10]. However, unlike these prior works, our construction is vulnerable to attacks when the depth of the computation graph exceeds an *a priori* fixed constant. See Remarks 2 and 3 for details.

(3) Efficient constructions of bounded-depth accumulation

We construct efficient bounded-depth accumulation schemes from any (non-homomorphic) vector commitment scheme (e.g., random-oracle based Merkle trees) and *any* linear code.

Compared to the prior state-of-the-art accumulation schemes, our construction has several advantages beyond just avoiding public-key assumptions, such as true linear time for the accumulation prover,[3] and plausible post-quantum security.[4] (We note that both our construction and prior accumulation schemes rely on the random oracle model; it is an open question to construct an accumulation scheme in the standard model.)

(4) Efficiency of and optimizations for instantiated PCD

Instantiating our generic PCD construction with our accumulation scheme leads to PCD schemes with numerous prover efficiency benefits compared to prior work. We also provide several optimizations for this scheme, including support for "batch" accumulation, a new low-overhead compiler from low-depth PCD to IVC, and a new *hybrid* PCD scheme that combines our low-depth PCD with any SNARK-based PCD scheme to achieve the best of both worlds. We detail the impact of these optimizations in Table 1.

Our scheme also has prover efficiency benefits beyond those captured by Table 1. To elaborate on these, we briefly detail the key factors that contribute to PCD prover cost. In all practical PCD constructions, to produce a proof for the next step of the distributed computation, the PCD prover invokes an underlying cryptographic proof system to prove satisfaction of a circuit C_{PCD} that contains a circuit representation C_Φ of the computation step (or *PCD predicate* Φ), and also performs other checks required by the PCD construction. Thus, PCD prover efficiency is determined by the prover efficiency of this proof system, and by the cost $|C_{\mathsf{PCD}}| - |C_\Phi|$ of the additional checks. We call the former the *proof-system overhead*, and the latter the PCD *circuit overhead*.

[3] To accumulate proofs for circuits of size n, prior accumulation schemes require performing an n-sized multi-scalar multiplication over elliptic-curve groups. As explained in prior work [37], this operation requires $\omega(n)$ group operations, and is hence not truly linear time.

[4] We only claim plausible post-quantum security, as we prove our construction in the random oracle model, instead of the quantum random oracle model [12]. We believe that our results can be extended to the latter model by applying techniques from prior work [25], but leave this to future work.

Table 1 Comparison of IVC schemes constructed from PCD over a tree of depth d and arity m. All costs omit constant factors. The circuit overhead measures $|C_{\mathsf{PCD}}| - |C_{\Phi}|$. The table displays the cost per invocation of Φ. Above n is the size of the recursive circuit, $n^{\star} = O(d^{\star}mn)$ is the circuit size of the accumulation decider for the recursive circuit, $|C_{\mathsf{MT}}|$ is the circuit size of checking a membership proof in a Merkle Tree with n leaves. T_{MT} is the time it takes to verify a Merkle Tree opening. Note that Fractal supports IVC computations of unbounded length, whereas all but the hybrid construction presented in this paper have a length bound of m^d.

scheme	circuit overhead per step	IVC verifier
Fractal [26]	$\lambda \log n \cdot \lvert C_{\mathsf{MT}}\rvert$	$\lambda \log n \cdot T_{\mathsf{MT}}$
+ STIR [2] (concurrent)	$(\log n + \lambda \log\log n) \cdot \lvert C_{\mathsf{MT}}\rvert$	$(\log n + \lambda \log\log n) \cdot T_{\mathsf{MT}}$
this paper	$d \cdot \lambda \cdot \lvert C_{\mathsf{MT}}\rvert$	$d \cdot n$
+ batch comm. (Sec 2.5)	$\frac{d\cdot\lambda}{m} \cdot \lvert C_{\mathsf{MT}}\rvert$	$d \cdot m \cdot n$
+ low-overhead IVC (Sec 2.6)	$\frac{d\cdot\lambda}{m^2} \cdot \lvert C_{\mathsf{MT}}\rvert$	$d \cdot m \cdot n$
+ hybrid (Sec 2.7)	$(\frac{d^{\star}\cdot\lambda}{m^2} + \frac{\lambda \log n^{\star}}{m^{d^{\star}}}) \cdot \lvert C_{\mathsf{MT}}\rvert$	$\lambda \log n^{\star} \cdot T_{\mathsf{MT}}$

PCD constructions based on ROM-based SNARKs have been able to achieve best-in-class proof-system overhead via extensive asymptotic and concrete optimizations (e.g., by achieving truly linear-time provers [15, 16, 17, 37], or relying on small fields [31]), but suffer from high PCD circuit overhead as the SNARK verifier subcircuit must perform numerous expensive random oracle calls.

Accumulation-based PCD constructions, on the other hand, have the smallest PCD circuit overhead (e.g., just ten thousand gates for Nova [41] vs. over one million gates for Fractal [26]). They also try to lower proof-system overhead by avoiding some cryptographic work (e.g., by requiring commitments only to the witness [20]), and via efficient arithmetizations of predicates (e.g., via cheap custom gates[5]). However, the remaining cryptographic work is quite expensive as it requires public-key operations, and hence the proof-system overhead of these constructions is relatively high.

Our construction of bounded-depth accumulation achieves the best of both worlds: it enjoys the low PCD circuit overhead of accumulation-based PCD constructions while taking advantage of the proof-system optimizations enjoyed by both approaches. In fact, the combination opens up new efficiency improvements that are not possible in either of the two paradigms alone. For example, all known ROM-SNARK-based PCD constructions thus far have relied on Reed–Solomon codes, and incur quasilinear prover costs from the quasilinear encoding time of these codes. Our construction, on the other hand, allows the use of any linear code, including linear-time-encodable codes [50, 33, 37], thus reducing prover costs.

1.2 Related work

PCD from symmetric-key assumptions

As noted in Section 1, the only end-to-end construction of PCD from symmetric-key assumptions is that of Chiesa, Ojha, and Spooner [26]. We provide a quantitative comparison in Table 1, and focus here on a qualitative comparison. Their construction is based on the Fractal SNARK, which they prove secure in the random oracle model.[6]

[5] Custom gates enforce checks beyond simple addition and multiplication. Examples include high-degree polynomial gates [35] and lookup gates [15, 36]. See [23, 30, 3] for empirical evidence of the benefits of such gates.

[6] Like us, their PCD construction must instantiate the random oracle with a concrete hash function, and can hence only achieve heuristic security.

Boneh, Drake, Fisch, and Gabizon [13] propose an optimization of the foregoing approach that batches the most expensive component, the low-degree test, across multiple proofs. While this concretely reduces prover cost, it does not lead to an asymptotic improvement in the prover overhead.

Like Fractal and similar SNARKs, our construction is able to take advantage of recent advances in the design of efficient code-based Interactive Oracle Proofs (IOPs) [7, 48]. For example, like recent work [51, 46, 31], we can greatly improve efficiency by relying on extension fields of small characteristic. Furthermore, unlike existing works, our fields do not need to have any special algebraic structure (e.g., large multiplicative subgroups).

PCD from public-key assumptions

Except the foregoing, all existing concretely-efficient IVC/PCD constructions [9, 18, 22, 21, 13, 41, 39, 20, 34, 40] rely on public-key assumptions, and in particular rely on the hardness of computing discrete logarithms over elliptic curve groups, which forces the usage of cryptographically large fields. Furthermore, efficient implementations require *cycles of elliptic curves*, which have proved unwieldy to implement correctly in practice [43]. In comparison, our construction avoids the need for public-key assumptions and this additional algebraic structure, and is able to use non-cryptographic field sizes.

Concurrent work

The concurrent work LatticeFold [11] takes a complementary approach to constructing plausibly post-quantum accumulation-based PCD: it constructs an accumulation/folding scheme from lattice-based assumptions. Unlike our construction, their accumulation scheme directly supports unbounded depth, allowing it to serve as a drop-in replacement in many existing constructions of accumulation-based PCD. However, this comes at a cost: at each accumulation step, the prover must demonstrate that the accumulation result has a small norm, which incurs a non-trivial computational cost. Furthermore, the dependence on lattice-based assumptions means that their scheme still relies on public-key assumptions.

An interesting direction for future work would be to combine the two approaches to reduce the need for small-norm checks. For example, one could construct a more efficient PCD scheme by first designing a bounded-depth accumulation scheme that relies on the bounded homomorphism supported by lattice commitments. One could then use the transformation of Section 2.7 to combine the resulting (bounded-depth) PCD scheme with the (unbounded-depth) LatticeFold PCD scheme to achieve a more efficient construction than either scheme alone.

Another concurrent work, STIR [2], provides a new proximity test that can be used as a drop-in replacement for the proximity test [5] used in Fractal [26]. As noted in Table 1, our schemes still have lower recursion overhead than this optimized baseline. Furthermore, this new STIR + Fractal-based PCD should also be compatible with our hybrid construction from Section 2.7.

▶ Remark 1 (security of bounded-depth PCD). All PCD schemes (including ours) only provably support computation graphs of depth $O(1)$. However, while there are no known attacks that break the security of prior schemes when the depth is $\omega(1)$, the same is not true for our scheme. As we explain in Section 2.1, our scheme is vulnerable to a relatively straightforward attack that obviates any security guarantees when the depth of the computation graph exceeds an *a priori* fixed constant. We emphasize that even such bounded-depth PCD is already powerful enough to support many interesting applications, including the primary application of constructing polynomial-length IVC [10]. See Remarks 2 and 3 for a more detailed discussion.

2 Techniques

We begin by reviewing the definition of an accumulation scheme [22, 21].[7] At a high level, it is used to perform *batch verification* of a predicate which, for us, will be a non-interactive argument's verifier $\mathcal{V}$. In other words, an accumulation scheme is used to check that $\mathcal{V}(\mathbb{x}_1, \pi_1), \ldots, \mathcal{V}(\mathbb{x}_n, \pi_n)$ all accept, more efficiently than the naive approach of individually verifying each instance $\mathbb{x}_i$ and proof π_i.

The workflow of an accumulation scheme is as follows. There are three main algorithms: a prover P, verifier V, and decider D. The prover is initialized with an empty accumulator acc_0, which is used to accumulate an input $(\mathbb{x}_1, \pi_1)$ into a new accumulator acc_1. The prover additionally outputs a proof; we write this as $(\mathsf{acc}_1, \mathsf{pf}_1) \leftarrow \mathrm{P}(\mathbb{x}_1, \pi_1, \mathsf{acc}_0)$. Later, acc_1 can be used to accumulate a second input, i.e., $(\mathsf{acc}_2, \mathsf{pf}_2) \leftarrow \mathrm{P}(\mathbb{x}_2, \pi_2, \mathsf{acc}_1)$, and so on. The correctness of a sequence of accumulations can then be established by checking that: (a) each accumulation step is valid, i.e., $\mathrm{V}(\mathbb{x}_i, \pi_i, \mathsf{acc}_{i-1}, \mathsf{acc}_i, \mathsf{pf}_i) = 1$; and (b) the final accumulator is valid, i.e., $\mathrm{D}(\mathsf{acc}_n) = 1$.

Notice that the decider only acts on the final accumulator, whereas the verifier acts on each accumulation step. Therefore, the crux of an accumulation scheme is making verification as cheap as possible. Towards this, we require that an accumulator acc can be split into a short instance part $\mathsf{acc}.\mathbb{x}$ and a (possibly) long witness part $\mathsf{acc}.\mathbb{w}$; we use $\mathsf{acc} = (\mathsf{acc}.\mathbb{x}, \mathsf{acc}.\mathbb{w})$ as shorthand. Similarly, we require that an argument proof can be split into instance and witness parts $\pi = (\pi.\mathbb{x}, \pi.\mathbb{w})$. The point is that the verifier can only look at the instance parts; we write this as $\mathrm{V}(\mathbb{x}_i, \pi_i.\mathbb{x}, \mathsf{acc}_{i-1}.\mathbb{x}, \mathsf{acc}_i.\mathbb{x}, \mathsf{pf}_i)$.

Definition

An accumulation scheme must satisfy the following properties.

- *Completeness:* The honest accumulation $(\mathsf{acc}', \mathsf{pf}) \leftarrow \mathrm{P}(\mathbb{x}, \pi, \mathsf{acc})$ of *any* valid input and accumulator should pass both the verifier's and decider's checks. That is, if $\mathcal{V}(\mathbb{x}, \pi) = 1$ and $\mathrm{D}(\mathsf{acc}) = 1$, then $\mathrm{V}(\mathbb{x}, \pi.\mathbb{x}, \mathsf{acc}.\mathbb{x}, \mathsf{acc}'.\mathbb{x}, \mathsf{pf}) = 1$ and $\mathrm{D}(\mathsf{acc}') = 1$.
- *Knowledge soundness:* If a new accumulator acc' passes the verifier's and decider's checks, then an efficient extractor can find a valid input and old accumulator that explains acc'. That is, if $\mathrm{D}(\mathsf{acc}') = 1$ and $\mathrm{V}(\mathbb{x}, \pi.\mathbb{x}, \mathsf{acc}.\mathbb{x}, \mathsf{acc}'.\mathbb{x}, \mathsf{pf}) = 1$, then an efficient extractor can find the witness part of the proof, $\pi.\mathbb{w}$, and the witness part of the old accumulator, $\mathsf{acc}.\mathbb{w}$, such that $\mathcal{V}(\mathbb{x}, \pi) = 1$ and $\mathrm{D}(\mathsf{acc}) = 1$.
- *Efficiency:* The cost of running the accumulation verifier n times plus the cost of running the accumulation decider once should be lower than the cost of running the argument verifier n times.

Accumulation schemes can be generalized to handle multiple inputs and accumulators in each step. For example, the prover's syntax would be $\mathrm{P}([\mathbb{x}_i, \pi_i]_{i=1}^{m_1}, [\mathsf{acc}_i]_{i=1}^{m_2})$, where m_1 and m_2 are the arities; see Section 3 for a comprehensive definition.

Prior constructions

All prior accumulation schemes [22, 21, 41, 20, 34, 39, 40] crucially use *additively homomorphic vector commitment schemes*. Informally, a vector commitment scheme allows one to construct a succinct commitment cm to a vector $\mathbf{v} \in \mathbb{F}^n$. The scheme is additively homomorphic if,

[7] We restrict our presentation to *split* accumulation schemes, as defined in [21].

given $\mathsf{cm}_1 = \mathsf{Commit}(\mathbf{v}_1)$ and $\mathsf{cm}_2 = \mathsf{Commit}(\mathbf{v}_2)$, $\mathsf{cm}_3 = \alpha \cdot \mathsf{cm}_1 + \beta \cdot \mathsf{cm}_2$ is a commitment to $\alpha\mathbf{v}_1 + \beta\mathbf{v}_2$. We remark that all known additively homomorphic vector commitment schemes, e.g., Pedersen commitments [45], rely on public-key assumptions.

The general blueprint for an accumulation scheme is as follows. An accumulator witness $\mathsf{acc}.\mathbb{w} \in \mathbb{F}^n$ is a vector, and the corresponding instance $\mathsf{acc}.\mathbb{x}$ is a commitment to $\mathsf{acc}.\mathbb{w}$. For simplicity, suppose the prover claims that acc_1 and acc_2 accumulate into acc_3. Roughly speaking, we want to guarantee that the output accumulator is a random linear combination of the input accumulators. The verifier checks this by computing the linear combination of the input commitments $\mathsf{acc}_1.\mathbb{x}$ and $\mathsf{acc}_2.\mathbb{x}$, and checking that the result equals the output commitment $\mathsf{acc}_3.\mathbb{x}$ [41, 20]. Later, the decider will check that $\mathsf{acc}_3.\mathbb{w}$ is a "good" vector, and that $\mathsf{acc}_3.\mathbb{x}$ commits to $\mathsf{acc}_3.\mathbb{w}$. Since commitments are binding, acc_3 must be the correct linear combination of acc_1 and acc_2.

We have omitted many details, most notably how to accumulate argument proofs. However, from this description alone we can observe two key properties of the vector commitment scheme. First, it has succinct commitments; this allows the verifier to be efficient. Second, it is additively homomorphic; this allows the verifier to perform meaningful checks. As noted earlier, this combination of properties unfortunately seems to require public-key assumptions.

2.1 Checking linearity

To overcome the foregoing limitation, we make a key observation: to verify that the output accumulator is a linear combination of the input accumulators, it is not necessary to directly compute a linear combination of the input commitments. Instead, it suffices to *check* that the output accumulator commits to a vector that is a linear combination of the vectors committed by the input accumulators. This idea is a natural one, and has appeared before under the name of "linear combination schemes" [13].

Recall that we want to check that $\mathsf{acc}_3.\mathbb{w} = \alpha \cdot \mathsf{acc}_1.\mathbb{w} + \beta \cdot \mathsf{acc}_2.\mathbb{w}$, where $\alpha, \beta \in \mathbb{F}$ are previously chosen scalars. More precisely, we want to check that $\mathbf{v}_3 = \alpha\mathbf{v}_1 + \beta\mathbf{v}_2$, where $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ are the underlying committed vectors of $\mathsf{acc}_1.\mathbb{x}$, $\mathsf{acc}_2.\mathbb{x}$, and $\mathsf{acc}_3.\mathbb{x}$ (and since commitments are binding, these vectors correspond with the accumulator witnesses). A *linearity check* is a protocol between the prover and the verifier that convinces the verifier of this claim. Assuming the verifier is public-coin, this can be made non-interactive using random oracles.

Our goal is to construct a linearity check which does not require homomorphic vector commitments. Instead, we require vector commitments with *local openings*. Informally, these allow the prover to generate a succinct proof that the underlying committed vector's i-th element is some claimed value. For example, Merkle tree commitments support local openings with proof size $O(\lambda \log n)$.

Distance spot checks

The key tool that we will be relying on is a protocol for convincing the verifier that two committed vectors are at most a constant distance apart. Concretely, let cm_1 and cm_2 be commitments to vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ respectively. We say that $\mathbf{v}_1$ and $\mathbf{v}_2$ are δ-far apart if they differ in at most δn locations. To show that $\mathbf{v}_1$ and $\mathbf{v}_2$ are at most δ-far apart, the prover and verifier engage in the following protocol:

1. The verifier uniformly samples an index $i \in [n]$ and sends it to the prover.
2. The prover responds with the purported i-th elements of $\mathbf{v}_1$ and $\mathbf{v}_2$, along with opening proofs.

3. The verifier accepts if these opening proofs are valid and the claimed elements are equal. Clearly, if $\mathbf{v}_1$ and $\mathbf{v}_2$ are δ-far apart, then the verifier will reject with probability δ. For any constant $\delta > 0$, this soundness error can be made negligible with $\Theta(\lambda)$ parallel repetitions.

Linearity spot checks

This protocol easily generalizes to testing any kind of element-wise property, and in particular we can use it to check that $\mathbf{v}_3$ is δ-close to the "virtual vector" $\alpha\mathbf{v}_1 + \beta\mathbf{v}_2$. Unfortunately, we need to ensure that the two vectors are equal at all locations. Suppose a cheating prover commits to a vector that only differs from $\alpha\mathbf{v}_1 + \beta\mathbf{v}_2$ at a *single* location j. Detecting this would require $\Theta(\lambda n)$ repetitions (essentially opening the entire commitment), which violates the accumulation verifier's efficiency requirement.

2.1.1 Error-resilient linearity checks from codes

It seems that we are at an impasse: our spot check can only guarantee that two vectors are δ-close, but the accumulation scheme requires exact agreement. To overcome this issue, we need to make our accumulation scheme resilient to a constant δ-fraction of corruptions. We do so by relying on linear codes, and in particular those which enjoy good distance properties, such as the Reed–Solomon code [47].

At a high level, we make the following changes to the accumulation scheme blueprint. Let C be a linear code, and let δ be a constant which is smaller than the unique decoding radius of C. The accumulator witness is a codeword $C(\mathbf{w})$, and the corresponding instance $\mathsf{acc}.\mathbb{x}$ is a commitment to $\mathsf{acc}.\mathbb{w}$. The accumulation verifier checks that the output accumulator is δ-close to a random linear combination of the input accumulators by running the linearity spot check. Later, the decider will check that $\mathsf{acc}_3.\mathbb{w}$ is the encoding of a good vector, and that $\mathsf{acc}_3.\mathbb{x}$ commits to $\mathsf{acc}_3.\mathbb{w}$.

Knowledge soundness

We would like our linearity spot check to satisfy the following knowledge soundness property. Suppose a (possibly malicious) prover outputs commitments cm_1, cm_2, and cm_3 which pass the check. Furthermore, suppose that cm_3 commits to a vector $\mathbf{v}_3$. Then an efficient extractor can find vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ such that (a) $\alpha\mathbf{v}_1 + \beta\mathbf{v}_2$ is δ-close to $\mathbf{v}_3$; and (b) cm_1 and cm_2 commit to $\mathbf{v}_1$ and $\mathbf{v}_2$. Notice that if we can extract vectors that satisfy 2.1.1, then our previous analysis of the spot check implies 2.1.1. Extraction turns out to be fairly straightforward: if we use Merkle commitments with a random oracle as the hash function, then we can find $\mathbf{v}_1$ and $\mathbf{v}_2$ by observing the prover's random oracle queries [52].[8]

Returning to accumulation, suppose that a (possibly malicious) prover outputs $\mathsf{acc}_1.\mathbb{x}$, $\mathsf{acc}_2.\mathbb{x}$, and acc_3 which pass the verifier's and decider's checks. Since the verifier runs the spot check, we can extract accumulator witnesses $\mathsf{acc}_1.\mathbb{w}$ and $\mathsf{acc}_2.\mathbb{w}$ such that $\alpha \cdot \mathsf{acc}_1.\mathbb{w} + \beta \cdot \mathsf{acc}_2.\mathbb{w}$ is δ-close to $\mathsf{acc}_3.\mathbb{w}$. Since the decider accepts $\mathsf{acc}_3.\mathbb{w}$, we know that $\mathsf{acc}_3.\mathbb{w}$ is a codeword $C(\mathbf{w}_3)$. Similarly, we need $\mathsf{acc}_1.\mathbb{w}$ and $\mathsf{acc}_2.\mathbb{w}$ to be codewords in order for the decider to accept $\mathsf{acc}_1.\mathbb{w}$ and $\mathsf{acc}_2.\mathbb{w}$. Unfortunately, this is simply not the case. For example, a cheating prover can always choose $\mathsf{acc}_1.\mathbb{w}$ which agrees with a codeword at all but one location, and this will almost certainly go undetected.

[8] To be precise, we must also consider a malicious prover that does not commit to a full vector; see the full version of the paper.

Can we still say something meaningful about the extracted witnesses? We argue that intuitively, since α and β are (possibly correlated) random scalars, with high probability $\mathsf{acc}_1.\mathbb{w}$ and $\mathsf{acc}_2.\mathbb{w}$ are themselves δ-close to codewords. Moreover, $\mathsf{acc}_1.\mathbb{w}$ and $\mathsf{acc}_2.\mathbb{w}$ decode to $\mathbf{w}_1$ and $\mathbf{w}_2$ such that $\alpha \cdot \mathbf{w}_1 + \beta \cdot \mathbf{w}_2 = \mathbf{w}_3$. This intuition can be formally proven using a suitable "proximity gap" result [6], which exists for a variety of parameter regimes.

The upshot is that we extract accumulators acc_1 and acc_2 which are only accepted by a *relaxed* decider. Namely, given an accumulator acc, this decider checks that $\mathsf{acc}.\mathbb{x}$ commits to $\mathsf{acc}.\mathbb{w}$, and moreover that $\mathsf{acc}.\mathbb{w}$ is δ-close to the code.

Recursive extraction

The foregoing analysis suffices for a single step of accumulation. However, in order to construct PCD, we will have to recursively extract from old accumulators. It is straightforward to see that a recursively extracted accumulator is only guaranteed to be 2δ-close to the code, since we are extracting from an accumulator that may already be δ-far from the code. More generally, k steps of recursion will only guarantee accumulators that are $k\delta$-close to the code. We will see that once $k\delta$ is larger than the unique decoding radius, extraction is no longer meaningful. In particular, this leads to the following concrete attack: a cheating prover can start with a bad codeword (rejected by the decider) and, over the k accumulation steps, incrementally move it to a good codeword (accepted by the decider). This motivates our notion of "bounded-depth" accumulation, which is not captured by existing definitions [21].

2.2 Defining bounded-depth accumulation

To describe our construction which only supports accumulation up to a certain (constant) depth, we introduce a new, relaxed knowledge soundness property; the key differences are highlighted in blue. We say that an accumulation scheme has *bounded-depth knowledge soundness* (with maximum depth d) if there exists a *family of deciders* $\{\mathrm{D}_s\}_{s=0}^{d}$, where D is equivalent to D_0, such that the following holds. If $\mathrm{D}_{s-1}(\mathsf{acc}') = 1$ and $\mathrm{V}(\mathbb{x}, \pi.\mathbb{x}, \mathsf{acc}.\mathbb{x}, \mathsf{acc}'.\mathbb{x}) = 1$, then an efficient extractor can find $\pi.\mathbb{w}$ and $\mathsf{acc}.\mathbb{w}$ such that $\mathcal{V}(\mathbb{x}, \pi) = 1$ and $\mathrm{D}_s(\mathsf{acc}) = 1$.

This is a meaningful definition. In addition to generalizing standard knowledge soundness, which can be recovered by setting $d = \infty$ and using a single decider D, it captures our construction based on error-resilient linearity checks: D_s is the decider that only accepts if the accumulator is at most $s\delta$-far from the code, and in particular D_0 only accepts codewords. The depth bound d is the maximum number of recursive extractions that we can perform before $d\delta$ exceeds the unique decoding radius of the code.

2.3 Bounded-depth PCD from bounded-depth accumulation

Existing theorems that build PCD from accumulation [21, 13, 41] do not immediately translate to the bounded-depth setting. To see why, let us recall a simplified version of the construction from [21]. Suppose we have an accumulation scheme for a *non-interactive argument of knowledge* (NARK). Informally, a PCD proof for z_i, which consists of a NARK proof π_i and accumulator acc_i, certifies that $z_i = F^i(z_0)$, where z_0 is some initial value. We maintain the invariant that if π_i and acc_i are valid, then the computation is correct up to the i-th step.

- The PCD prover receives a proof (π_i, acc_i) for z_i, and wants to output a proof for z_{i+1}. First, it accumulates π_i and acc_i into a new accumulator acc_{i+1}, generating an accumulation proof pf_{i+1}. Next, it generates a NARK proof π_{i+1} for the following claim, expressed as a circuit R (see Figure 1): "$z_{i+1} = F(z_i)$, and there exists a NARK proof π_i, old accumulator acc_i, and accumulation proof pf_{i+1} which correctly accumulate into acc_{i+1}." The proof for z_{i+1} is $(\pi_{i+1}, \mathsf{acc}_{i+1})$.

$R(\mathbb{x} = (z_{i+1}, \mathsf{acc}_{i+1}.\mathbb{x}), \mathbb{w} = (z_i, \pi_i.\mathbb{x}, \mathsf{acc}_i.\mathbb{x}, \mathsf{pf}_{i+1})) :$
1. Check that $z_{i+1} = F(z_i)$.
2. Set $\mathbb{x}_i = (z_i, \mathsf{acc}_i.\mathbb{x})$.
3. Check that $\mathrm{V}(\mathbb{x}_i, \pi_i, \mathsf{acc}_i.\mathbb{x}, \mathsf{acc}_{i+1}.\mathbb{x}, \mathsf{pf}_{i+1}) = 1$.

Figure 1 Recursion circuit for PCD.

- The PCD verifier checks a proof (π_i, acc_i) for z_i by running the NARK verifier on π_i and the decider on acc_i.

Now suppose we replace the accumulation scheme with one that only has bounded-depth knowledge soundness. The construction remains the same, but we must provide a new soundness analysis.

PCD knowledge soundness

We need to construct an extractor which, given an accepting proof (π_T, acc_T) for z_T, extracts a sequence of values $z_0, \ldots, z_T$ such that $z_{i+1} = F(z_i)$ for all i. [21] gives the following strategy, which *interleaves* the NARK extractor and accumulation extractor. Suppose we have z_{i+1}, π_{i+1}, and acc_{i+1}. First, we invoke the NARK extractor to obtain $(z_i, \pi_i.\mathbb{x}, \mathsf{acc}_i.\mathbb{x}, \mathsf{pf}_{i+1})$. Second, we invoke the accumulation extractor to obtain $(\pi_i.\mathbb{w}, \mathsf{acc}_i.\mathbb{w})$. This gives us π_i and acc_i, and the process continues. We maintain the invariant that in the i-th step, π_i and acc_i are valid.

With bounded-depth accumulation, we need to maintain a slightly weaker invariant: in the i-th step, instead of requiring that acc_i is accepted by the *strict* decider D, we only require that it is accepted by the i-th *relaxed* decider D_i. This discussion only provides a high-level overview of the proof strategy, and only describes an *IVC* construction; we describe the full PCD construction that supports arbitrary (bounded-depth) computation graphs, along with a full soundness analysis, in the full version of the paper.

▶ Remark 2 (bounded-depth PCD suffices). As presented, our PCD scheme supports up to d steps of computation, where d is the maximum depth of the accumulation scheme. We call this *bounded-depth PCD*. Since d will realistically be a small constant, this seems to be of limited use: most computations require more than a constant number of steps! Fortunately, even such a limited PCD scheme can be used to construct IVC for *any* polynomial-length computation [10]. The idea is for the PCD prover to receive multiple proofs in each step, yielding a *computation tree*. In particular, if we can accumulate m inputs and m accumulators in a single step, then our PCD scheme can support computation trees of size m^d. Setting $m = \lambda$ and $d = O(1)$ allows us to support polynomial-size computations.

▶ Remark 3 (bounded-depth vs. constant-depth PCD). Perhaps surprisingly, even with standard (unbounded) accumulation, [21] is only able to construct *constant-depth PCD*.[9] This is because the size of the PCD extractor grows exponentially in the computation's depth, regardless of the accumulation scheme's knowledge soundness property. We remark that this limitation is largely theoretical: there is no known attack which exploits unbounded recursive proof composition. In contrast, the depth bound in bounded-depth PCD is not merely an artifact of the analysis: there exists a concrete attack that can be mounted against our construction when the depth exceeds d. This means that the tree-based strategy described in Remark 2 is *necessary* for real-world implementations, unlike in prior work.

[9] This is slightly different from bounded-depth PCD, where the maximum depth must be fixed in advance.

2.4 Constructing bounded-depth accumulation

Our starting point is the ProtoStar and ProtoGalaxy accumulation schemes [20, 34]. They support a general class of non-interactive arguments where consists of the NP witness along with a short vector commitment to the witness, and whose verifier performs three steps: (a) computing Fiat–Shamir challenges; (b) checking vector commitment openings; and (c) evaluating a polynomial over the instance, challenges, and openings. The key insight of ProtoStar and ProtoGalaxy is that the accumulation verifier can cheaply perform the first step, while batching the remaining steps and deferring it to the decider.

Let us recall ProtoGalaxy's strategy [34], focusing on the case where we are trying to accumulate claims about satisfaction of R1CS, a popular NP language in the succinct argument literature. Recall that an R1CS instance consists of constraint matrices $A, B, C \in \mathbb{F}^{N \times (\ell+n)}$, and is said to be satisfied by a public input $x \in \mathbb{F}^\ell$ and witness $w \in \mathbb{F}^n$ if $A \cdot (x||w) \circ B \cdot (x||w) - C \cdot (x||w) = 0^N$. In ProtoGalaxy, a NARK proof for an R1CS instance with public input x would consist of the proof witness, which is just the R1CS witness w, and the proof instance, which is just a vector commitment to w. The polynomial p evaluated by the verifier is of the form $p(x, w) = A \cdot (x||w) \circ B \cdot (x||w) - C \cdot (x||w)$. Then, to accumulate R1CS claims, ProtoGalaxy's accumulator witness $\mathsf{acc.w}$ is *also* a vector $w \in \mathbb{F}^n$, and its accumulator instance $\mathsf{acc.x}$ consists of a vector $\vec{x} \in \mathbb{F}^\ell$, a (homomorphic) vector commitment to w, and an error term $e \in \mathbb{F}$. Thus, a NARK proof verification claim can be seen as a special case of the accumulator verification case where the error term is zero.

To accumulate m such (accumulation or NARK) verification claims for accumulators $\mathsf{acc}_1, \ldots, \mathsf{acc}_m$, ProtoGalaxy relies on the following univariate polynomial identity:

$$p\left(\sum_{i=1}^{m} L_i(X) \cdot x_i, \sum_{i=1}^{m} L_i(X) \cdot w_i\right) = \sum_{i=1}^{m} L_i(X) \cdot e_i \mod v_H(X),$$

where L_i is the i-th Lagrange polynomial of some m-sized set $H \subset \mathbb{F}$, and $v_H(X)$ is the vanishing polynomial on H. For some quotient polynomial q, this claim can be rewritten as

$$p\left(\sum_{i=1}^{m} L_i(X) \cdot x_i, \sum_{i=1}^{m} L_i(X) \cdot w_i\right) = \sum_{i=1}^{m} L_i(X) \cdot e_i + q(X) \cdot v_H(X),$$

and this formulation allows the verifier to check the identity at a random point $\alpha \in \mathbb{F}$ if the prover sends q.

This allows us to accumulate the m input claims as follows. The prover constructs a new accumulator acc whose new instance is $x := \sum_i L_i(\alpha) \cdot x_i$, new witness vector is $w := \sum_i L_i(\alpha) \cdot w_i$, and the new error is $e := v(\alpha) \cdot q(\alpha) + \sum_i L_i(\alpha) \cdot e_i$. The verifier checks that acc was computed correctly by homomorphically computing the new vector commitment, and directly computing the new instance and new error term. Finally, the accumulation decider completes the checks by asserting that that $p(x, w) = e$.[10]

Our construction

We follow the same strategy, except we replace homomorphic computations with error-resilient linearity checks. Concretely, we make the following changes. An accumulator witness is a codeword $f \in C$, and an accumulator instance consists of a vector commitment to f and an

[10] Technically speaking, this check, as stated, is ill-formed, as the output of p is an N-sized vector, while e is a single field element. However, this can be fixed by taking a random linear combinations of the output vector. ProtoStar and ProtoGalaxy show how to represent the latter task *also* as a low-degree polynomial evaluation.

error term $e \in \mathbb{F}$. The decider accepts an accumulator if f is the encoding of a vector $w \in \mathbb{F}^n$ such that $p(x, w) = e$. Finally, as discussed in Section 2.1, the verifier uses a linearity check to ensure that the new accumulator acc is sufficiently consistent with the old accumulators $\mathsf{acc}_1, \ldots, \mathsf{acc}_m$.

Overall, our construction inherits many desirable properties from ProtoStar [20] and ProtoGalaxy [34], including support for arbitrary arity $m = \text{poly}(\lambda)$, which is crucial for constructing PCD (see Remark 2), and efficient support for custom gates.

Extension to arbitrary linear codes

A key parameter in our construction is the linear code. We use Reed–Solomon codes because they display a proximity gap when the coefficients are Lagrange evaluations $L_1(\alpha), \ldots, L_m(\alpha)$. However, a modified version of our construction extends to arbitrary linear codes. This is advantageous as these codes can have asymptotically faster encoding time compared to Reed–Solomon codes [37]. The construction can work with any proximity gap (e.g., uniformly random coefficients). The key idea is to commit to two codes within the accumulator, one for proximity checking and the other for the algebraic check using $p(X)$. See Section 4.3 for details.

Efficiency

The cost of the accumulation verifier is dominated by that of the linearity checker. Recall that to achieve negligible knowledge soundness error at depth d, the latter checks that two code words are $d \cdot \delta$ close via $k = O(d \cdot \lambda)$ spot checks, where δ is such that $d \cdot \delta$ is less than the unique decoding radius of the code. Overall, when instantiated with the Merkle tree-based vector commitment, the accumulation verifier checks $O(d \cdot \lambda)$ Merkle tree paths. When applying this construction to the PCD scheme in Section 2.3, the latter cost becomes the *recursive overhead* of the PCD prover. As noted in Table 1, this cost is asymptotically better than the $O(\log n \cdot \lambda)$ cost of prior SNARK-based PCD schemes [26].

A keen reader may notice that a disadvantage of our construction is that recursive overhead scales with the depth of the PCD computation graph. We now present several optimizations that reduce the depth of the PCD tree and significantly improve the efficiency of the resulting PCD scheme in practice.

2.5 Optimization: batch commitments

Recall from Remarks 2 and 3 that for our PCD construction, reducing the depth of the computation graph is essential for achieving provable security guarantees, and the standard depth-reduction technique for the case of IVC [10] works by constructing a *PCD tree* whose leaves comprise the actual computation being performed. To achieve constant computation depth, Bitansky et al. [10] set the arity m of this tree to be super-constant (i.e., $m = \lambda$). The per-node recursive overhead of our PCD construction in this setting is the cost of performing linearity checks on m codewords of size n, which costs $O(m \cdot \lambda \cdot \log n)$ hashes when using Merkle trees. We now describe how to reduce this to just $O(\lambda(\log n + m))$ hashes.

Recall that the linearity checker opens all m codewords at the same locations. This means that for each spot-check, each of the m Merkle trees is opened at the same leaf. We take advantage of this repetitive structure by committing to all m codewords using a *single* Merkle tree whose i-th leaf is the concatenation of the i-th symbols of the codewords. Each spot-check now requires opening only a single Merkle tree path (and checking the leaf hash), leading to a cost of $O(\lambda(\log n + m))$ hashes for $O(\lambda)$ spot checks, as required.

However, this modification comes at a cost: it requires us to commit to all codewords together at the same time. While this is straightforward at the leaf layer, it gets more complex at higher layers. For example, even committing to a new accumulator now requires waiting for the batch of "sibling" new accumulators, which in turn means that we must wait for m accumulations (each of size m) to complete before we can compute the commitments to the m new accumulators. Overall, across the entire tree, this requires the PCD prover to maintain a state of $O(m^2)$ "pending" accumulators. We provide more details in the full version of the paper.

2.6 Optimization: low-overhead IVC from accumulation

We give a generic optimization which improves on the PCD-to-IVC compiler of Bitansky et al. [52, 10]. They construct a (polynomial-length) IVC scheme for a function F by using a (constant-depth) PCD scheme for a related function F'. In particular, F' computes F (at leaf nodes) and performs consistency checks (at internal nodes). This is wasteful: even though we do not need to prove anything about F at internal nodes, the PCD prover still generates a proof for F' (which is dominated by F).

We improve this compiler by constructing an IVC scheme with minimal overhead. The core idea is that we first construct a tree of accumulators for F, i.e., a tree the leaf nodes are proofs for F, and each parent accumulates its children. Then, we construct a PCD tree which proves that the accumulation tree was constructed correctly. When the PCD scheme is instantiated with our accumulation-based construction, the PCD circuit now checks *two* accumulation verifiers: one that checks the correctness of the accumulation tree, and one that helps check the correctness of the PCD tree.

Accumulating separately means that we no longer have to generate NARK proofs for F at internal nodes. Additionally, because we only need to show that internal nodes of the accumulation tree were constructed correctly, our PCD tree has one fewer layer than before. This further reduces cost, in particular for higher arities (as m grows, the leaf layer of a tree contains a higher fraction of nodes). We provide more details in the full version of this paper.

2.7 Optimization: PCD composition

The recursive overhead of our PCD scheme grows linearly with the maximum supported depth. This is contrast with SNARK-based PCD schemes like Fractal [26], which do not suffer from such an efficiency loss. However, these schemes pay a higher per-step cost anyway, and are thus asymptotically less efficient than our PCD scheme for low recursion depths.

We provide a generic optimization to combine SNARK-based PCD schemes with our PCD scheme to achieve a scheme that achieves better efficiency than either scheme alone. The core idea is to first use our accumulation-based PCD up to some depth d_1, and then prove the PCD verifier for the latter with a SNARK-based PCD scheme on top. When invoked with tree PCD, this means that the SNARK-based PCD scheme is invoked only every m^{d_1} steps. By choosing d_1 appropriately, the resulting scheme achieves better efficiency than either scheme alone. Furthermore, the resulting scheme supports *arbitrary* constant depth, as opposed to our accumulation-based scheme, which only supports an *a priori* fixed depth. We provide more details in the full version of this paper.

3 Bounded-depth accumulation

Let $\mathsf{ARG} = (\mathcal{G}, \mathcal{P}, \mathcal{V})$ be a non-interactive argument where knowledge soundness (but not necessarily completeness) additionally holds for a "relaxed" verifier $\tilde{\mathcal{V}}$. Suppose proofs can be split into two parts, i.e., $\pi = (\pi.\mathbb{x}, \pi.\mathbb{w})$. Let qx denote $(\mathbb{x}, \pi.\mathbb{x})$, where $\mathbb{x}$ is an instance of the relation; call this the *verifier input instance*. Let qw denote $\pi.\mathbb{w}$; call this the *verifier input witness*. We write $\mathcal{V}(\mathsf{pp}, \mathbb{i}, \mathsf{qx}_i, \mathsf{qw}_i)$ as shorthand for the verifier running on $\mathbb{x}$ and π. We write acc as shorthand for the tuple $(\mathsf{acc}.\mathbb{x}, \mathsf{acc}.\mathbb{w})$. An *accumulation scheme* in the random oracle model for ARG is a tuple of algorithms $\mathsf{AS} = (\mathrm{G}, \mathrm{I}, \mathrm{P}, \mathrm{V}, \mathrm{D})$ with the following syntax and properties.

- $\mathrm{G}(1^\lambda) \to \mathsf{pp}_{\mathsf{AS}}$: On input a security parameter λ, the generator G samples and outputs accumulation parameters $\mathsf{pp}_{\mathsf{AS}}$.
- $\mathrm{I}(\mathsf{pp}_{\mathsf{AS}}, \mathsf{pp}, \mathbb{i}) \to (\mathsf{apk}, \mathsf{avk}, \mathsf{dk})$: On input accumulation parameters $\mathsf{pp}_{\mathsf{AS}}$ and argument parameters pp, the indexer I deterministically computes and outputs a proving key apk, verification key avk, and decision key dk.
- $\mathrm{P}(\mathsf{apk}, [\mathsf{qx}_i, \mathsf{qw}_i]_{i=1}^{m_1}, [\mathsf{acc}_i]_{i=1}^{m_2}) \to (\mathsf{acc}, \mathsf{pf})$: On input the proving key apk, verifier inputs $[\mathsf{qx}_i, \mathsf{qw}_i]_{i=1}^{m_1}$, and accumulators $[\mathsf{acc}_i]_{i=1}^{m_2}$, the accumulation prover P outputs a new accumulator acc and proof pf. Here, m_1 and m_2 are fixed arities which may be functions of λ.
- $\mathrm{V}(\mathsf{avk}, [\mathsf{qx}_i]_{i=1}^{m_1}, [\mathsf{acc}.\mathbb{x}_i]_{i=1}^{m_2}, \mathsf{acc}.\mathbb{x}, \mathsf{pf}) \to \{0,1\}$: On input the verifying key avk, verifier input instances $[\mathsf{qx}_i]_{i=1}^{m_1}$, accumulator instances $[\mathsf{acc}.\mathbb{x}_i]_{i=1}^{m_2}$, new accumulator instance $\mathsf{acc}.\mathbb{x}$, and proof pf, the accumulation verifier V outputs a bit indicating whether $\mathsf{acc}.\mathbb{x}$ correctly accumulates the other instances.
- $\mathrm{D}(\mathsf{dk}, \mathsf{acc}) \to \{0,1\}$: On input the decision key dk and accumulator acc, the decider outputs a bit indicating whether acc is a valid accumulator.

Completeness

For every (unbounded) adversary $\mathcal{A}$,

$$\Pr\left[\begin{array}{c} \forall i \in [m_1], \mathcal{V}(\mathsf{pp}, \mathbb{i}, \mathsf{qx}_i, \mathsf{qw}_i) = 1 \\ \forall i \in [m_2], \mathrm{D}(\mathsf{dk}, \mathsf{acc}_i) = 1 \\ \Downarrow \\ \mathrm{V}(\mathsf{avk}, [\mathsf{qx}_i]_{i=1}^{m_1}, [\mathsf{acc}_i.\mathbb{x}]_{i=1}^{m_2}, \mathsf{acc}.\mathbb{x}, \mathsf{pf}) = 1 \\ \mathrm{D}(\mathsf{dk}, \mathsf{acc}) = 1 \end{array} \middle| \begin{array}{r} \rho \leftarrow \mathcal{U}(\lambda) \\ \mathsf{pp}_{\mathsf{AS}} \leftarrow \mathrm{G}(1^\lambda) \\ \mathsf{pp} \leftarrow \mathcal{G}(1^\lambda) \\ (\mathbb{i}, [\mathsf{qx}_i, \mathsf{qw}_i]_{i=1}^{m_1}, [\mathsf{acc}_i]_{i=1}^{m_2}) \leftarrow \mathcal{A}(\mathsf{pp}_{\mathsf{AS}}, \mathsf{pp}) \\ (\mathsf{apk}, \mathsf{avk}, \mathsf{dk}) \leftarrow \mathrm{I}(\mathsf{pp}_{\mathsf{AS}}, \mathsf{pp}, \mathbb{i}) \\ (\mathsf{acc}, \mathsf{pf}) \leftarrow \mathrm{P}(\mathsf{apk}, [\mathsf{qx}_i, \mathsf{qw}_i]_{i=1}^{m_1}, [\mathsf{acc}_i]_{i=1}^{m_2}) \end{array}\right] = 1.$$

To bootstrap accumulation, we also assume the prover can efficiently construct a dummy accumulator and proof, denoted $\mathsf{acc} = \mathrm{P}(\mathsf{apk}, \bot)$, which the decider accepts.

Knowledge soundness

We say that AS has bounded-depth knowledge soundness (with maximum depth d_s) if there exists a *family of deciders* $\{\mathrm{D}_s\}_{s=0}^{d_s}$, where D is equivalent to D_0, such that the following holds. There exists an expected polynomial time extractor E such that for every depth parameter $s \in [d_s]$, expected polynomial time adversary $\tilde{\mathrm{P}}$, and auxiliary input distribution $\mathcal{D}$, the following probability is negligibly close to 1:

$$\Pr\left[\begin{array}{c} \mathrm{V}^\rho(\mathsf{avk}, [\mathsf{qx}_i]_{i=1}^{m_1}, [\mathsf{acc}_i.\mathbb{x}]_{i=1}^{m_2}, \mathsf{acc}.\mathbb{x}, \mathsf{pf}) = 1 \\ \mathrm{D}_{s-1}(\mathsf{dk}, \mathsf{acc}) = 1 \\ \Downarrow \\ \forall i \in [m_1], \tilde{\mathcal{V}}(\mathsf{pp}, \mathbb{i}, \mathsf{qx}_i, \mathsf{qw}_i) = 1 \\ \forall i \in [m_2], \mathrm{D}_s(\mathsf{dk}, \mathsf{acc}_i) = 1 \end{array} \middle| \begin{array}{r} \rho \leftarrow \mathcal{U}(\lambda) \\ \mathsf{pp}_{\mathsf{AS}} \leftarrow \mathrm{G}(1^\lambda) \\ \mathsf{pp} \leftarrow \mathcal{G}(1^\lambda) \\ \mathsf{ai} \leftarrow \mathcal{D}(1^\lambda) \\ (\mathbb{i}, [\mathsf{qx}_i]_{i=1}^{m_1}, [\mathsf{acc}_i.\mathbb{x}]_{i=1}^{m_2}, \mathsf{acc}, \mathsf{pf}; r) \leftarrow \tilde{\mathrm{P}}(\mathsf{pp}_{\mathsf{AS}}, \mathsf{pp}, \mathsf{ai}) \\ ([\mathsf{qw}_i]_{i=1}^{m_1}, [\mathsf{acc}_i.\mathbb{w}]_{i=1}^{m_2}) \leftarrow \mathrm{E}^{\tilde{\mathrm{P}}}(\mathsf{pp}_{\mathsf{AS}}, \mathsf{pp}, \mathsf{ai}, r) \\ (\mathsf{apk}, \mathsf{avk}, \mathsf{dk}) \leftarrow \mathrm{I}(\mathsf{pp}_{\mathsf{AS}}, \mathsf{pp}, \mathbb{i}) \end{array}\right]$$

The extractor implicitly receives s as input.

▶ Remark 4. We can also define a bounded-depth version of completeness with a separate family of deciders. In the completeness definition, valid accumulators for the i-th decider accumulate to a valid accumulator for the $(i+1)$-th decider. This is important in settings where even an honest prover can only perform a bounded number of accumulations.

4 Constructing bounded-depth accumulation

We construct a bounded-depth accumulation scheme, which supports (possibly distinct) arities $m_1 = \text{poly}(\lambda)$ and $m_2 = \text{poly}(\lambda)$, for a general class of non-interactive arguments. Across all of our constructions, we fix the following global constants: depth bound d_s, code rate $R \in (0,1)$, and distance $\delta \leq (1-R)/(2d_s)$; this guarantees that $d_s\delta$ is smaller than the unique decoding radius of a Reed–Solomon code with rate R. Additionally, we use domain separation on the random oracle ρ to model three disjoint oracles: ρ_H for Merkle trees and hashing, ρ_{ARG} for the argument verifier's randomness, and ρ_{AS} for the accumulation verifier's randomness. When querying ρ_{ARG} and ρ_{AS}, we assume the random oracle's output is used to sample from the verifier's challenge set. All security proofs are deferred to the full version of the paper.

Notation

If x, y, z are vectors, we will often interpret tuples like $(x,(y,z))$ as a vector consisting of x, y, z concatenated together. Given a codeword $f \in C$, let $\hat{f}$ denote the corresponding message which encodes to f. We write $\mathbf{f}$ as shorthand for the tuple $(f^{(1)}, \ldots, f^{(\mu)})$ and $\mathbf{C}$ as shorthand for the Cartesian product $C^{(1)} \times \cdots \times C^{(\mu)}$. Similarly, let $\hat{\mathbf{f}} = (\hat{f}^{(1)}, \ldots, \hat{f}^{(\mu)})$ and $\Delta(\mathbf{f}, \mathbf{g}) = \max_{j \in [\mu]} \Delta(f^{(j)}, g^{(j)})$. When querying locations in a codeword, let $\mathbf{Q} = (\mathcal{Q}^{(1)}, \ldots, \mathcal{Q}^{(\mu)})$ and $\mathbf{f}[\mathbf{Q}] = (f^{(1)}[\mathcal{Q}^{(1)}], \ldots, f^{(\mu)}[\mathcal{Q}^{(\mu)}])$. We use arrow notation as shorthand for tuples of commitment data, e.g., $\vec{\mathsf{cm}} = (\mathsf{cm}^{(1)}, \ldots, \mathsf{cm}^{(\mu)})$. Vector commitment functions map over tuples, e.g., $(\vec{\mathsf{cm}}, \vec{\mathsf{aux}}) \leftarrow \mathsf{VC.Commit}(\mathsf{vp}, \mathbf{f})$ should be interpreted as saying "for each $j \in [\mu]$, let $(\vec{\mathsf{cm}}^{(j)}, \vec{\mathsf{aux}}^{(j)}) \leftarrow \mathsf{VC.Commit}(\mathsf{vp}, f^{(j)})$."

4.1 Non-interactive argument

Our starting point is any special-sound interactive proof with an algebraic verifier. By this, we mean that the verifier's check can be expressed as a sequence of degree d polynomials (derived from the index) which take the transcript as input. The verifier accepts if all of the polynomials evaluate to zero.

In more detail, we require an interactive proof for some indexed relation $\mathcal{R}(\mathbb{F})$. Let μ be the number of rounds in the protocol; this may be a function of the index. For simplicity, we assume that the instance $\mathbb{x}$, the prover's messages $m^{(1)}, \ldots, m^{(\mu)}$, and the verifier's challenges $r^{(1)}, \ldots, r^{(\mu)}$ are all vectors over $\mathbb{F}$; their lengths may be a function of the index. From the index, the verifier derives degree d polynomials $p_1, \ldots, p_\ell$ over $\mathbb{F}$. It accepts if, for all i, $p_i(\mathbb{x}, \mathbf{r}, \mathbf{m}) = 0$.

Compressing the verifier

Without loss of generality, we can assume that the algebraic verifier consists of a single polynomial. This is because multiple polynomial checks can be compressed into a single check, e.g., by using the following technique due to [34, 20].

Let Π be an interactive proof where the verifier's check consists of ℓ polynomials $p_0, \ldots, p_{\ell-1}$, each of degree d. We transform this into an interactive proof $\mathsf{CV}[\Pi]$ for the same relation, where the verifier's check is a single polynomial

$$p(\vec{X}, \vec{Y}) = \sum_{i=0}^{\ell-1} \mathsf{pow}_i(\vec{Y}) \cdot p_i(\vec{X}).$$

Here, pow_i is the unique degree $\log \ell$ polynomial satisfying $\mathsf{pow}_i(1, \beta, \beta^2, \beta^4, \ldots, \beta^{\ell/2}) = \beta^i$ for all $\beta \in \mathbb{F}$. It follows that p is of degree $d + \log \ell$. The interactive protocol is the same as before, except that the verifier additionally samples a challenge $y = (1, \beta, \beta^2, \beta^4, \ldots, \beta^{\ell/2})$ where $\beta \leftarrow \mathbb{F}$. The verifier accepts if $p(x, y) = 0$, where x is the transcript from the original proof.

If Π is $(k_1, \ldots, k_\mu)$-special-sound, then $\mathsf{CV}[\Pi]$ is $(k_1, \ldots, k_\mu, \ell)$-special-sound. To see why, suppose we have a tree T of accepting transcripts for $\mathsf{CV}[\Pi]$. Consider an arbitrary node in the penultimate layer of the tree. Its children correspond with transcripts of the form $(x, y_1), \ldots, (x, y_\ell)$, each y_i distinct. Since the transcripts are accepting, the degree $\ell - 1$ univariate polynomial $\sum_{i=0}^{\ell-1} Z^i \cdot p_i(x)$ is zero at ℓ points. It follows that f is the zero polynomial and, for all i, $p_i(x) = 0$. Let T' denote T with its bottom layer removed. We have shown that T' is a tree of accepting transcripts for Π. Hence, we construct an extractor for $\mathsf{CV}[\Pi]$ by running the extractor for Π on T'.

Committing to messages

In order to achieve efficient accumulation, we will instead have the prover send commitments to messages. Only in the final move of the protocol does the prover send openings to all of the commitments. Strictly speaking, this is only special-sound for an "augmented relation"; namely, there exists an extractor which, given a tree of accepting transcripts, outputs either a witness or a break of the commitment scheme. Nonetheless, assuming the scheme is computationally binding, applying the Fiat-Shamir transformation yields a non-interactive argument of knowledge for the original relation. We refer to [20] for a more detailed analysis.

Removing interaction

Given a special-sound interactive proof (P, V), we apply the Fiat–Shamir transformation (with commitments) to get a non-interactive argument of knowledge [4]. In order to achieve efficient accumulation, we use a standard variant of the transformation where the index is first hashed to a succinct value τ. The Fiat–Shamir transform outputs a non-interactive argument $\mathsf{ARG} = (\mathcal{G}, \mathcal{P}, \mathcal{V})$, shown in Figure 2, for the indexed relation family $\{\mathcal{R}_{\mathsf{pp}} = \mathcal{R}(\mathbb{F})\}$. We also define an indexer $\mathcal{I}$ as a helper algorithm.

Attema et al. [4] prove that multi-round public-coin protocols can be compiled into non-interactive arguments. However, their definition of knowledge-soundness is slightly different from ours: they require that the extractor succeeds with non-negligible probability if the adversary succeeds with non-negligible probability. Our definition, on the other hand, requires that the extractor succeeds with all but negligible probability whenever the adversary succeeds. However, these definitions are equivalent as one can boost the extractor's success probability by running it until it succeeds. This works as the extractor is only required to run in *expected* polynomial time.

$\mathcal{G}(1^\lambda)$:

1. Choose a suitable field $\mathbb{F}, \log|\mathbb{F}| = \Omega(\lambda)$.
2. Let $\mathsf{vp} \leftarrow \mathsf{VC.Setup}(1^\lambda, \mathbb{F})$.
3. Output $\mathsf{pp} = (\mathbb{F}, \mathsf{vp})$.

$\mathcal{I}^\rho(\mathsf{pp} = (\mathbb{F}, \mathsf{vp}), \mathbb{i})$:

1. Query $\tau \leftarrow \rho_H(\mathbb{i})$.
2. From $\mathbb{F}$ and $\mathbb{i}$, derive the following parameters, collected into $\mathbb{p}$:
   - The number of rounds, denoted μ.
   - The length of the instance.
   - For each $j \in [\mu]$, the length, denoted $\ell^{(j)}$, of the prover's j-th message.
   - For each $j \in [\mu]$, the format of the verifier's j-th challenge.
3. From $\mathbb{F}$ and $\mathbb{i}$, derive the verifier's check p.
4. For each $j \in [\mu]$, let $C^{(j)}$ be a Reed–Solomon code over $\mathbb{F}$ with dimension $\ell^{(j)}$, rate R, and evaluation domain $L^{(j)}$.
5. Output $\mathsf{ipk} = (\mathbb{F}, \mathsf{vp}, \mathbb{i}, \tau, \mathbb{p}, \mathbf{C})$ and $\mathsf{ivk} = (\mathbb{F}, \mathsf{vp}, \tau, \mathbb{p}, p, \mathbf{C})$.

$\mathcal{P}^\rho(\mathsf{pp}, \mathbb{i}, \mathbb{x}, \mathbb{w})$:

1. Compute the proving key $\mathsf{ipk} = (\mathbb{F}, \mathsf{vp}, \mathbb{i}, \tau, \mathbb{p}, \mathbf{C})$ according to $\mathcal{I}^\rho(\mathsf{pp}, \mathbb{i})$.
2. For $j \in [\mu]$:
   - Compute the prover's j-th message $m^{(j)} \leftarrow P(\mathbb{F}, \mathbb{i}, \mathbb{x}, \mathbb{w}, [m^{(i)}, r^{(i)}]_{i=1}^{j-1})$.
   - Encode $m^{(j)}$ to $f^{(j)} \in C^{(j)}$.
   - Let $(\mathsf{cm}^{(j)}, \mathsf{aux}^{(j)}) \leftarrow \mathsf{VC.Commit}^{\rho_H}(\mathsf{vp}, f^{(j)})$.
   - Query $r^{(j)} \leftarrow \rho_{\mathsf{ARG}}(\tau, \mathbb{x}, \mathsf{cm}^{(1)}, \ldots, \mathsf{cm}^{(j)})$.
3. Output $\pi = (\vec{\mathsf{cm}}, \vec{\mathsf{aux}})$.

$\mathcal{V}^\rho(\mathsf{pp}, \mathbb{i}, \mathbb{x}, \pi = (\vec{\mathsf{cm}}, \vec{\mathsf{aux}}))$:

1. Compute the verification key $\mathsf{ivk} = (\mathbb{F}, \mathsf{vp}, \tau, \mathbb{p}, p, \mathbf{C})$ according to $\mathcal{I}^\rho(\mathsf{pp}, \mathbb{i})$.
2. For each $j \in [\mu]$, query $r^{(j)} \leftarrow \rho_{\mathsf{ARG}}(\tau, \mathbb{x}, \mathsf{cm}^{(1)}, \ldots, \mathsf{cm}^{(j)})$.
3. Let $\mathbf{f} \leftarrow \mathsf{VC.Answer}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{cm}}, \vec{\mathsf{aux}})$.
4. Verify that $\mathbf{f} \in \mathbf{C}$.
5. Accept if $p(\mathbb{x}, \mathbf{r}, \hat{\mathbf{f}}) = 0$.

$\tilde{\mathcal{V}}^\rho(\mathsf{pp}, \mathbb{i}, \mathbb{x}, \pi = (\vec{\mathsf{cm}}, \vec{\mathsf{aux}}))$:

1. Compute the verification key $\mathsf{ivk} = (\mathbb{F}, \mathsf{vp}, \tau, \mathbb{p}, p, \mathbf{C})$ according to $\mathcal{I}^\rho(\mathsf{pp}, \mathbb{i})$.
2. For each $j \in [\mu]$, query $r^{(j)} \leftarrow \rho_{\mathsf{ARG}}(\tau, \mathbb{x}, \mathsf{cm}^{(1)}, \ldots, \mathsf{cm}^{(j)})$.
3. Let $\mathbf{f} \leftarrow \mathsf{VC.Answer}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{cm}}, \vec{\mathsf{aux}})$.
4. Find $\mathbf{g} \in \mathbf{C}$ by uniquely decoding $\mathbf{f}$. If no such codeword exists, reject.
5. Accept if $p(\mathbb{x}, \mathbf{r}, \hat{\mathbf{g}}) = 0$.

Figure 2 Non-interactive argument algorithms.

Relaxed verifier

We use a specific commitment scheme to support accumulation: the prover sends a vector commitment to a codeword of the message, along with the full auxiliary data. We relax the verifier in two different ways to get $\tilde{\mathcal{V}}$ (also shown in Figure 2). First, we allow it to decode noisy codewords. Second, we allow it to accept partial openings to the vector commitment where the missing positions are interpreted as erasure errors. Because the verifier only accepts when the partial opening is decodable, the prover is still bound to a unique (decoded) vector; hence, knowledge soundness is not affected.

4.2 Accumulation scheme

Fix a subset $H \subset \mathbb{F}$ of size $m = m_1 + m_2$; this may be a function of λ. We construct an accumulation scheme $\mathsf{AS} = (\mathrm{G}, \mathrm{I}, \mathrm{P}, \mathrm{V}, \mathrm{D})$ for ARG. The argument verifier inputs are split into $\mathsf{qx} = (\mathbb{x}, \vec{\mathsf{cm}})$ and $\mathsf{qw} = \mathsf{a}\vec{\mathsf{u}}\mathsf{x}$. Accumulators have a nearly identical structure; in fact, any verifier input $(\mathsf{qx}, \mathsf{qw})$ can be converted into an accumulator acc by setting $\mathsf{acc}.\mathbb{w} = \mathsf{qw}$ and $\mathsf{acc}.\mathbb{x} = \textsc{CastInput}^{\rho}(\tau, \mathsf{qx})$, which is defined below.

$\textsc{CastInput}^{\rho}(\tau, \mathsf{qx} = (\mathbb{x}, \vec{\mathsf{cm}}))$:
1. For each $j \in [\mu]$, query $r^{(j)} \leftarrow \rho_{\mathsf{ARG}}(\tau, \mathbb{x}, \mathsf{cm}^{(1)}, \ldots, \mathsf{cm}^{(j)})$.
2. Collect $\mathbb{x}$ and $\mathbf{r}$ into a vector x.
3. Output $\mathsf{acc}.\mathbb{x} = (0, x, \vec{\mathsf{cm}})$.

We also define a helper function which casts $[\mathsf{qx}_i]_{i=1}^{m_1}$ and $[\mathsf{acc}_i.\mathbb{x}]_{i=1}^{m_2}$ into a single list of $m = m_1 + m_2$ accumulator instances.

$\textsc{Cast}^{\rho}(\tau, [\mathsf{qx}_i]_{i=1}^{m_1}, [\mathsf{acc}_i.\mathbb{x}]_{i=1}^{m_2})$:
1. Output $[\textsc{CastInput}(\tau, \mathsf{qx}_1), \ldots, \textsc{CastInput}(\tau, \mathsf{qx}_{m_1}), \mathsf{acc}_1.\mathbb{x}, \ldots, \mathsf{acc}_{m_2}.\mathbb{x}]$.

Finally, we define helper functions which perform the bulk of proving and verification.

$\textsc{Prove}^{\rho}(\mathsf{apk}, [e_i, x_i, \vec{\mathsf{cm}}_i, \mathsf{a}\vec{\mathsf{u}}\mathsf{x}_i]_{i=1}^{m})$:
1. For each $i \in [m]$, let $\mathbf{f}_i \leftarrow \mathsf{VC.Answer}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{cm}}_i, \mathsf{a}\vec{\mathsf{u}}\mathsf{x}_i)$.
2. Compute the univariate polynomial $q \in \mathbb{F}[X]$ of degree at most $d(m-1) - m$ such that
$$p\left(\sum_{i=1}^{m} L_{i,H}(X) \cdot (x_i, \hat{\mathbf{f}}_i)\right) = v_H(X) \cdot q(X) + \sum_{i=1}^{m} L_{i,H}(X) \cdot e_i.$$
3. Query $\alpha \leftarrow \rho_{\mathsf{AS}}(\tau, [\mathsf{acc}_i.\mathbb{x}]_{i=1}^{m}, q)$, where $\alpha \in \mathbb{F}$ is a uniformly sampled field element.
4. Compute $e = v(\alpha) \cdot q(\alpha) + \sum_i L_i(\alpha) \cdot e_i$.
5. Compute $(x, \hat{\mathbf{f}}) = \sum_i L_i(\alpha) \cdot (x_i, \hat{\mathbf{f}}_i)$.
6. Let $(\vec{\mathsf{cm}}, \mathsf{a}\vec{\mathsf{u}}\mathsf{x}) \leftarrow \mathsf{VC.Commit}^{\rho_H}(\mathsf{vp}, \mathbf{f})$.
7. Set $\mathsf{acc}.\mathbb{x} = (e, x, \vec{\mathsf{cm}})$ and $\mathsf{acc}.\mathbb{w} = \mathsf{a}\vec{\mathsf{u}}\mathsf{x}$.
8. Query $\mathbf{Q} \leftarrow \rho_{\mathsf{AS}}(\tau, [\mathsf{acc}_i.\mathbb{x}]_{i=1}^{m}, q, \mathsf{acc}.\mathbb{x})$, where $\mathcal{Q}^{(j)}$ is a uniformly sampled t-sized subset of $L^{(j)}$.
9. For each $i \in [m]$, let $\vec{\mathsf{op}}_i \leftarrow \mathsf{VC.Open}^{\rho_H}(\mathsf{vp}, \mathsf{a}\vec{\mathsf{u}}\mathsf{x}_i, \mathbf{Q})$.
10. Let $\vec{\mathsf{op}} \leftarrow \mathsf{VC.Open}^{\rho_H}(\mathsf{vp}, \mathsf{a}\vec{\mathsf{u}}\mathsf{x}, \mathbf{Q})$.
11. Output acc and $\mathsf{pf} = (q, [\vec{\mathsf{op}}_i]_{i=1}^{m}, \vec{\mathsf{op}})$.

$\textsc{Verify}^{\rho}(\mathsf{avk}, [e_i, x_i, \vec{\mathsf{cm}}_i]_{i=1}^m, (e, x, \vec{\mathsf{cm}}), (q, [\vec{\mathsf{op}}_i]_{i=1}^m, \vec{\mathsf{op}}))$:

1. Query $\alpha \leftarrow \rho_{\mathsf{AS}}(\tau, [\mathsf{acc}_i.\mathbb{x}]_{i=1}^m, q)$.
2. Query $\mathbf{Q} \leftarrow \rho_{\mathsf{AS}}(\tau, [\mathsf{acc}_i.\mathbb{x}]_{i=1}^m, q, \mathsf{acc}.\mathbb{x})$.
3. For each $i \in [m]$, let $\mathbf{v}_i = \mathsf{VC.Answer}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{cm}}_i, \vec{\mathsf{op}}_i)$. If $\mathbf{v}_i[\mathbf{Q}]$ contains $\perp$, reject.
4. Let $\mathbf{v} = \mathsf{VC.Answer}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{cm}}, \vec{\mathsf{op}}, \mathbf{Q})$. If $\mathbf{v}[\mathbf{Q}]$ contains $\perp$, reject.
5. Verify that $e = v_H(\alpha) \cdot q(\alpha) + \sum_{i=1}^m L_{i,H}(\alpha) \cdot e_i$.
6. Verify that $(x, \mathbf{v}) = \sum_{i=1}^m L_{i,H}(\alpha) \cdot (x_i, \mathbf{v}_i)$.

See Figure 3 for a full description of AS, along with the family of deciders.

$\mathrm{G}(1^\lambda)$:

1. Choose a suitable spot check parameter $t = \Omega(\lambda)$.
2. Output $\mathsf{pp}_{\mathsf{AS}} = t$.

$\mathrm{I}^{\rho}(\mathsf{pp}_{\mathsf{AS}} = t, \mathsf{pp} = (\mathbb{F}, \mathsf{vp}), \mathbb{i})$:

1. Obtain $\tau, \mathbb{p}, p, \mathbf{C}$ from $\mathcal{I}^{\rho}(\mathsf{pp}, \mathbb{i})$.
2. Output $\mathsf{apk} = (\mathbb{F}, \mathsf{vp}, t, \tau, \mathbb{p}, p, \mathbf{C})$, $\mathsf{avk} = (\mathbb{F}, \mathsf{vp}, t, \tau, \mathbb{p}, \mathbf{C})$, and $\mathsf{dk} = (\mathbb{F}, \mathsf{vp}, \mathbb{p}, p, \mathbf{C})$.

$\mathrm{P}^{\rho}(\mathsf{apk}, [\mathsf{qx}_i, \mathsf{qw}_i]_{i=1}^{m_1}, [\mathsf{acc}]_{i=1}^{m_2})$:

1. Let $[e_i, x_i, \vec{\mathsf{cm}}_i]_{i=1}^m \leftarrow \textsc{Cast}(\tau, [\mathsf{qx}_i]_{i=1}^{m_1}, [\mathsf{acc}_i.\mathbb{x}]_{i=1}^{m_2})$.
2. Parse $[\vec{\mathsf{aux}}_i]_{i=1}^m = [\mathsf{qw}_1, \ldots, \mathsf{qw}_{m_1}, \mathsf{acc}_1.\mathbb{w}, \ldots, \mathsf{acc}_{m_2}.\mathbb{w}]$.
3. Output $\textsc{Prove}^{\rho}(\mathsf{apk}, [e_i, x_i, \vec{\mathsf{cm}}_i, \vec{\mathsf{aux}}_i]_{i=1}^m)$.

$\mathrm{V}^{\rho}(\mathsf{avk}, [\mathsf{qx}_i]_{i=1}^{m_1}, [\mathsf{acc}_i.\mathbb{x}]_{i=1}^{m_2}, \mathsf{acc}.\mathbb{x}, \mathsf{pf})$:

1. Let $[e_i, x_i, \vec{\mathsf{cm}}_i]_{i=1}^m \leftarrow \textsc{Cast}(\tau, [\mathsf{qx}_i]_{i=1}^{m_1}, [\mathsf{acc}_i.\mathbb{x}]_{i=1}^{m_2})$.
2. Accept if $\textsc{Verify}(\mathsf{avk}, [e_i, x_i, \vec{\mathsf{cm}}_i]_{i=1}^m, \mathsf{acc}.\mathbb{x}, \mathsf{pf})$ accepts.

$\mathrm{D}_s^{\rho}(\mathsf{dk}, \mathsf{acc}.\mathbb{x} = (e, x, \vec{\mathsf{cm}}), \mathsf{acc}.\mathbb{w} = \vec{\mathsf{aux}})$:

1. Let $\mathbf{f} \leftarrow \mathsf{VC.Answer}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{cm}}, \vec{\mathsf{aux}})$.
2. Find $\mathbf{g} \in \mathbf{C}, \Delta(\mathbf{f}, \mathbf{g}) \leq s\delta$ by decoding $\mathbf{f}$. If no such codeword exists, reject.
3. Accept if $p(x, \hat{\mathbf{g}}) = e$.

Figure 3 Accumulation scheme algorithms.

4.3 Using arbitrary linear codes

At first glance, our accumulation scheme does not use any special properties of the Reed–Solomon code. Because other codes might have desirable properties, e.g., linear-time encoding, this motivates the following question: can we instantiate it with an arbitrary linear code, so long as it has good distance? Recall that the accumulator's vectors are a random linear combination of the input vectors. In particular, the coefficients are Lagrange evaluations of a random field element; this is necessary for compressing the polynomial evaluation claims. Unfortunately, existing proximity gaps for arbitrary linear codes [49, 1, 32] do not support this specific distribution of coefficients.

Separating the proximity claim

In some sense, our construction accumulates two distinct claims for the same vector: a polynomial evaluation claim, and a proximity claim. To overcome the foregoing issue, we modify our accumulation scheme to separate these claims. Specifically, each accumulator now

holds *two* codewords: the first is a linear combination using Lagrange coefficients, and the second is a linear combination using proximity gap coefficients. Accordingly, the accumulation verifier uses the first codeword to compress evaluation claims, and the second codeword to maintain proximity. This will be roughly twice as expensive as before.

▶ **Definition 5** (Proximity gaps for linear codes)**.** *Let C be a linear code with relative distance d and blocklength n. We will treat elements of C as functions $v : [n] \to \mathbb{F}$. Let* Gen *be a function that takes in randomness* rnd *and outputs coefficients $r_1, \ldots, r_m \in \mathbb{F}$. We say that C has a proximity gap with respect to distribution* Gen*, error* err*, and proximity bound* B *if the following holds.*

For any $\delta \leq \mathsf{B}$ and arbitrary functions $u_1, \ldots, u_m \in \mathbb{F}^n : [n] \to \mathbb{F}$, if

$$\Pr_{\mathsf{rnd}}\left[\Delta\left(\sum_{i=1}^m r_i \cdot u_i, C\right) \leq \delta \;:\; (r_1, \ldots, r_m) \leftarrow \mathsf{Gen}(\mathsf{rnd})\right] > \mathsf{err}(m),$$

then for all $(r_1, \ldots, r_m)$ in the support of the distribution $\mathsf{Gen}(\mathsf{rnd})$*, we have*

$$\Delta\left(\sum_{i=1}^m r_i \cdot u_i, C\right) \leq \delta.$$

Furthermore, there exists $v_1, \ldots, v_m \in C$ such that for all $(r_1, \ldots, r_m)$ in the support,

$$\Delta\left(\sum_{i=1}^m r_i \cdot u_i, \sum_{i=1}^m r_i \cdot v_i\right) \leq \delta,$$

and in fact $|\{x \in [n] : (u_1(x), \ldots, u_m(x)) \neq (v_1(x), \ldots, v_m(x))\}| \leq \delta n$.

▶ **Lemma 6.** *Let C be a linear code with relative distance d, blocklength n, and proximity gap with respect to distribution* Gen*, error* err*, and proximity bound* B*. For $\delta \leq \mathsf{B}$ and vectors $f_1, \ldots, f_m \in \mathbb{F}^n$, suppose the following holds:*

$$\Pr_{\mathsf{rnd}}\left[\Delta\left(\sum_{i=1}^m r_i \cdot f_i, C\right) \leq \delta \;:\; (r_1, \ldots, r_m) \leftarrow \mathsf{Gen}(\mathsf{rnd})\right] > \mathsf{err}(m)$$

Then there exists a subdomain $L' \subseteq [n]$ and codewords $g_1, \ldots, g_m \in C$ such that the following holds. First, $|L'|/[n] \geq 1 - \delta$. Second, for all $i \in [m]$, f_i and g_i agree on L'.

Construction

We use linear codes with efficient unique decoding up to $\mathsf{B} = O(1)$, the error bound of the proximity gap. Fix $\delta \leq \mathsf{B}/d_s$; this guarantees that $d_s\delta \leq \mathsf{B}$ which is smaller than the unique decoding radius. The modified construction is shown below; all changes are highlighted in blue. We omit the description of unchanged algorithms from Figure 3.

$\textsc{CastInput}^{\rho}(\tau, \mathsf{qx} = (\mathbb{x}, \vec{\mathsf{cm}}))$:

1. For each $j \in [\mu]$, query $r^{(j)} \leftarrow \rho_{\mathsf{ARG}}(\tau, \mathbb{x}, \mathsf{cm}^{(1)}, \ldots, \mathsf{cm}^{(j)})$.
2. Collect $\mathbb{x}$ and $\mathbf{r}$ into a vector x.
3. Output $\mathsf{acc}.\mathbb{x} = (0, x, \vec{\mathsf{cm}}, \vec{\mathsf{cm}}')$, where $\vec{\mathsf{cm}}'$ are commitments to zero vectors.

$\textsc{Prove}^{\rho}(\mathsf{apk}, [e_i, x_i, \vec{\mathsf{cm}}_i, \vec{\mathsf{aux}}_i, \vec{\mathsf{cm}}'_i, \vec{\mathsf{aux}}'_i]_{i=1}^m)$:

1. For each $i \in [m]$, let $\mathbf{f}_i \leftarrow \mathsf{VC.Answer}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{cm}}_i, \vec{\mathsf{aux}}_i)$, and $\mathbf{f}'_i \leftarrow \mathsf{VC.Answer}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{cm}}'_i, \vec{\mathsf{aux}}'_i)$.
2. Compute the univariate polynomial $q \in \mathbb{F}[X]$ of degree at most $d(m-1) - m$ such that

$$p\left(\sum_{i=1}^{m} L_{i,H}(X) \cdot (x_i, \hat{\mathbf{f}}_i)\right) = v_H(X) \cdot q(X) + \sum_{i=1}^{n} L_{i,H}(X) \cdot e_i.$$

3. Query $\alpha, \mathsf{rnd} \leftarrow \rho_{\mathsf{AS}}(\tau, [\mathsf{acc}_i.\mathbb{x}]_{i=1}^m, q)$, where $\alpha \in \mathbb{F}$ and rnd are sampled uniformly.
4. Compute $e = v(\alpha) \cdot q(\alpha) + \sum_i L_i(\alpha) \cdot e_i$.
5. Compute $(x, \hat{\mathbf{f}}) = \sum_i L_i(\alpha) \cdot (x_i, \hat{\mathbf{f}}_i)$.
6. Let $(r_1, \ldots, r_m, r'_1, \ldots, r'_m) \leftarrow \mathsf{Gen}(\mathsf{rnd})$. Compute $\mathbf{f}' = \sum_{i=1}^m r_i \cdot \mathbf{f}_i + r'_i \cdot \mathbf{f}'_i$.
7. Let $(\vec{\mathsf{cm}}, \vec{\mathsf{aux}}) \leftarrow \mathsf{VC.Commit}^{\rho_H}(\mathsf{vp}, \mathbf{f})$, and $(\vec{\mathsf{cm}}', \vec{\mathsf{aux}}') \leftarrow \mathsf{VC.Commit}^{\rho_H}(\mathsf{vp}, \mathbf{f}')$.
8. Set $\mathsf{acc}.\mathbb{x} = (e, x, \vec{\mathsf{cm}}, \vec{\mathsf{cm}}')$ and $\mathsf{acc}.\mathbb{w} = (\vec{\mathsf{aux}}, \vec{\mathsf{aux}}')$.
9. Query $\mathbf{Q} \leftarrow \rho_{\mathsf{AS}}(\tau, [\mathsf{acc}_i.\mathbb{x}]_{i=1}^m, q, \mathsf{acc}.\mathbb{x})$, where $\mathcal{Q}^{(j)}$ is a uniformly sampled t-sized subset of $[n]$.
10. For each $i \in [m]$, let $\vec{\mathsf{op}}_i \leftarrow \mathsf{VC.Open}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{aux}}_i, \mathbf{Q})$, and $\vec{\mathsf{op}}'_i \leftarrow \mathsf{VC.Open}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{aux}}'_i, \mathbf{Q})$.
11. Let $\vec{\mathsf{op}} \leftarrow \mathsf{VC.Open}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{aux}}, \mathbf{Q})$, and $\vec{\mathsf{op}}' \leftarrow \mathsf{VC.Open}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{aux}}', \mathbf{Q})$.
12. Output acc and $\mathsf{pf} = (q, [\vec{\mathsf{op}}_i, \vec{\mathsf{op}}'_i]_{i=1}^m, \vec{\mathsf{op}}, \vec{\mathsf{op}}')$.

$\textsc{Verify}^{\rho}(\mathsf{avk}, [e_i, x_i, \vec{\mathsf{cm}}_i, \vec{\mathsf{cm}}'_i]_{i=1}^m, (e, x, \vec{\mathsf{cm}}, \vec{\mathsf{cm}}'), (q, [\vec{\mathsf{op}}_i, \vec{\mathsf{op}}'_i]_{i=1}^m, \vec{\mathsf{op}}, \vec{\mathsf{op}}'))$:

1. Query $\alpha, \mathsf{rnd} \leftarrow \rho_{\mathsf{AS}}(\tau, [\mathsf{acc}_i.\mathbb{x}]_{i=1}^m, q)$.
2. Query $\mathbf{Q} \leftarrow \rho_{\mathsf{AS}}(\tau, [\mathsf{acc}_i.\mathbb{x}]_{i=1}^m, q, \mathsf{acc}.\mathbb{x})$.
3. For each $i \in [m]$, let $\mathbf{v}_i = \mathsf{VC.Answer}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{cm}}_i, \vec{\mathsf{op}}_i)$. If $\mathbf{v}_i[\mathbf{Q}]$ contains $\perp$, reject.
4. Let $\mathbf{v} = \mathsf{VC.Answer}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{cm}}, \vec{\mathsf{op}}, \mathbf{Q})$. If $\mathbf{v}[\mathbf{Q}]$ contains $\perp$, reject.
5. For each $i \in [m]$, let $\mathbf{v}'_i = \mathsf{VC.Answer}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{cm}}'_i, \vec{\mathsf{op}}'_i)$. If $\mathbf{v}'_i[\mathbf{Q}]$ contains $\perp$, reject.
6. Let $\mathbf{v}' = \mathsf{VC.Answer}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{cm}}', \vec{\mathsf{op}}', \mathbf{Q})$. If $\mathbf{v}'[\mathbf{Q}]$ contains $\perp$, reject.
7. Verify that $e = v_H(\alpha) \cdot q(\alpha) + \sum_{i=1}^n L_{i,H}(\alpha) \cdot e_i$.
8. Verify that $(x, \mathbf{v}) = \sum_{i=1}^m L_{i,H}(\alpha) \cdot (x_i, \mathbf{v}_i)$.
9. Let $(r_1, \ldots, r_m, r'_1, \ldots, r'_m) \leftarrow \mathsf{Gen}(\mathsf{rnd})$. Verify that $\mathbf{v}' = \sum_{i=1}^m r_i \cdot v_i + r'_i \cdot v'_i$.

$\mathrm{D}_s^{\rho}(\mathsf{dk}, \mathsf{acc}.\mathbb{x} = (e, x, \vec{\mathsf{cm}}, \vec{\mathsf{cm}}'), \mathsf{acc}.\mathbb{w} = (\vec{\mathsf{aux}}, \vec{\mathsf{aux}}'))$:

1. Let $\mathbf{f} \leftarrow \mathsf{VC.Answer}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{cm}}, \vec{\mathsf{aux}})$.
2. Let $\mathbf{f}' \leftarrow \mathsf{VC.Answer}^{\rho_H}(\mathsf{vp}, \vec{\mathsf{cm}}', \vec{\mathsf{aux}}')$.
3. Find $\mathbf{g}, \mathbf{g}' \in \mathbf{C}$ such that $\Delta(\mathbf{f}, \mathbf{g}), \Delta(\mathbf{f}', \mathbf{g}') \leq (s-1)\delta$ by decoding $\mathbf{f}$ and $\mathbf{f}'$. If no such codewords exists, reject.
4. Accept if $p(x, \hat{\mathbf{g}}) = e$.

References

1 Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In *Proceedings of the 24th ACM Conference on Computer and Communications Security*, CCS '17, pages 2087–2104, 2017. URL: `https://eprint.iacr.org/2022/1608`, `doi:10.1145/3133956.3134104`.

2 Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. STIR: reed-solomon proximity testing with fewer queries. In *Proceedings of the 44th Annual International Cryptology Conference*, CRYPTO '24, pages 380–413, 2024. URL: `https://eprint.iacr.org/2024/390`, `doi:10.1007/978-3-031-68403-6_12`.

3 Arasu Arun, Srinath T. V. Setty, and Justin Thaler. Jolt: Snarks for virtual machines via lookups. In *Proceedings of the 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '24, pages 3–33, 2024. URL: https://eprint.iacr.org/2023/1217, doi:10.1007/978-3-031-58751-1_1.

4 Thomas Attema, Serge Fehr, and Michael Klooß. Fiat-shamir transformation of multi-round interactive proofs (extended version). *J. Cryptol.*, 36(4):36, 2023. URL: https://eprint.iacr.org/2021/1377, doi:10.1007/S00145-023-09478-Y.

5 Eli Ben-Sasson, Iddo Bentov, Ynon Horesh, and Michael Riabzev. Fast Reed–Solomon interactive oracle proofs of proximity. In *Proceedings of the 45th International Colloquium on Automata, Languages and Programming*, ICALP '18, pages 14:1–14:17, 2018. doi:10.4230/LIPICS.ICALP.2018.14.

6 Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity gaps for reed-solomon codes. *JACM*, 70(5):31:1–31:57, 2023. URL: https://eprint.iacr.org/2020/654, doi:10.1145/3614423.

7 Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In *Proceedings of the 14th Theory of Cryptography Conference*, TCC '16-B, pages 31–60, 2016. URL: https://eprint.iacr.org/2016/116, doi:10.1007/978-3-662-53644-5_2.

8 Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Scalable zero knowledge via cycles of elliptic curves. In *Proceedings of the 34th Annual International Cryptology Conference*, CRYPTO '14, pages 276–294, 2014. URL: https://eprint.iacr.org/2014/595, doi:10.1007/978-3-662-44381-1_16.

9 Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Scalable zero knowledge via cycles of elliptic curves. *Algorithmica*, 79(4):1102–1160, 2017. URL: https://eprint.iacr.org/2014/595, doi:10.1007/S00453-016-0221-0.

10 Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. Recursive composition and bootstrapping for SNARKs and proof-carrying data. In *Proceedings of the 45th ACM Symposium on the Theory of Computing*, STOC '13, pages 111–120, 2013. URL: http://eprint.iacr.org/2012/095, doi:10.1145/2488608.2488623.

11 Dan Boneh and Binyi Chen. Latticefold: A lattice-based folding scheme and its applications to succinct proof systems. ePrint Report 2024/257, 2024. URL: https://eprint.iacr.org/2024/257.

12 Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *Proceedings of the 17th International Conference on the Theory and Application of Cryptology and Information Security*, ASIACRYPT '11, pages 41–69, 2011. URL: https://eprint.iacr.org/2010/428, doi:10.1007/978-3-642-25385-0_3.

13 Dan Boneh, Justin Drake, Ben Fisch, and Ariel Gabizon. Halo infinite: Proof-carrying data from additive polynomial commitments. In *Proceedings of the 41st Annual International Cryptology Conference*, CRYPTO '21, 2021. URL: https://eprint.iacr.org/2020/1536.

14 Joseph Bonneau, Izaak Meckler, Vanishree Rao, and Evan Shapiro. Coda: Decentralized cryptocurrency at scale. ePrint Report 2020/352, 2020. URL: https://eprint.iacr.org/2020/352.

15 Jonathan Bootle, Andrea Cerulli, Jens Groth, Sune K. Jakobsen, and Mary Maller. Arya: Nearly linear-time zero-knowledge proofs for correct program execution. In *Proceedings of the 24th International Conference on the Theory and Application of Cryptology and Information Security*, ASIACRYPT '18, pages 595–626, 2018. URL: https://eprint.iacr.org/2018/380, doi:10.1007/978-3-030-03326-2_20.

16 Jonathan Bootle, Alessandro Chiesa, and Jens Groth. Linear-time arguments with sublinear verification from tensor codes. In *Proceedings of the 18th International Conference on the Theory of Cryptography*, TCC '20, pages 19–46, 2020. URL: https://eprint.iacr.org/2020/1426, doi:10.1007/978-3-030-64378-2_2.

17 Jonathan Bootle, Alessandro Chiesa, and Siqi Liu. Zero-knowledge iops with linear-time prover and polylogarithmic-time verifier. In *Proceedings of the 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '22, pages 275–304, 2022. URL: https://eprint.iacr.org/2020/1527, doi:10.1007/978-3-031-07085-3_10.

18 Sean Bowe, Jack Grigg, and Daira Hopwood. Halo: Recursive proof composition without a trusted setup. ePrint Report 2019/1021, 2019. URL: https://eprint.iacr.org/2019/1021.

19 Elette Boyle, Ran Cohen, and Aarushi Goel. Breaking the $o(\sqrt{n})$-bit barrier: Byzantine agreement with polylog bits per party. *J. Cryptol.*, 37(1):2, 2024. arXiv:2002.02516.

20 Benedikt Bünz and Binyi Chen. Protostar: Generic efficient accumulation/folding for special-sound protocols. In *Proceedings of the 29th International Conference on the Theory and Application of Cryptology and Information Security*, ASIACRYPT '23, pages 77–110, 2023. URL: https://eprint.iacr.org/2023/620, doi:10.1007/978-981-99-8724-5_3.

21 Benedikt Bünz, Alessandro Chiesa, William Lin, Pratyush Mishra, and Nicholas Spooner. Proof-carrying data without succinct arguments. In *Proceedings of the 41st Annual International Cryptology Conference*, CRYPTO '21, pages 681–710, 2021. URL: https://eprint.iacr.org/2020/1618, doi:10.1007/978-3-030-84242-0_24.

22 Benedikt Bünz, Alessandro Chiesa, Pratyush Mishra, and Nicholas Spooner. Proof-carrying data from accumulation schemes. In *Proceedings of the 18th Theory of Cryptography Conference*, TCC '20, 2020. URL: https://eprint.iacr.org/2020/499.

23 Binyi Chen, Benedikt Bünz, Dan Boneh, and Zhenfei Zhang. Hyperplonk: Plonk with linear-time prover and high-degree custom gates. In *Proceedings of the 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '23, pages 499–530, 2023. URL: https://eprint.iacr.org/2022/1355, doi:10.1007/978-3-031-30617-4_17.

24 Weikeng Chen, Alessandro Chiesa, Emma Dauterman, and Nicholas P. Ward. Reducing participation costs via incremental verification for ledger systems. ePrint Report 2020/1522, 2020. URL: https://eprint.iacr.org/2020/1522.

25 Alessandro Chiesa, Peter Manohar, and Nicholas Spooner. Succinct arguments in the quantum random oracle model. In *Proceedings of the 17th International Conference on the Theory of Cryptography*, TCC '19, pages 1–29, 2019. URL: https://eprint.iacr.org/2019/834, doi:10.1007/978-3-030-36033-7_1.

26 Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. Fractal: Post-quantum and transparent recursive proofs from holography. In *Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '20, 2020. URL: https://eprint.iacr.org/2019/1076.

27 Alessandro Chiesa and Eran Tromer. Proof-carrying data and hearsay arguments from signature cards. In *Proceedings of the 1st Symposium on Innovations in Computer Science*, ICS '10, pages 310–331, 2010. URL: https://people.eecs.berkeley.edu/~alexch/docs/CT10.pdf.

28 Alessandro Chiesa, Eran Tromer, and Madars Virza. Cluster computing in zero knowledge. In *Proceedings of the 34th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT '15, pages 371–403, 2015. URL: https://eprint.iacr.org/2015/377, doi:10.1007/978-3-662-46803-6_13.

29 Stephen Chong, Eran Tromer, and Jeffrey A. Vaughan. Enforcing language semantics using proof-carrying data. ePrint Report 2013/513, 2013. URL: http://eprint.iacr.org/2013/513.

30 Michel Dellepere, Pratyush Mishra, and Alireza Shirzad. Garuda and pari: Faster and smaller snarks via equifficient polynomial commitments. ePrint Report 2024/1245, 2024. URL: https://eprint.iacr.org/2024/1245.

31 Benjamin E. Diamond and Jim Posen. Succinct arguments over towers of binary fields. ePrint Report 2023/1784, 2023. URL: https://eprint.iacr.org/2023/1784.

32 Benjamin E. Diamond and Jim Posen. Proximity testing with logarithmic randomness. *IACR Commun. Cryptol.*, 1(1):2, 2024. URL: https://eprint.iacr.org/2023/630, doi:10.62056/AKSDKP10.

33 Erez Druk and Yuval Ishai. Linear-time encodable codes meeting the gilbert-varshamov bound and their cryptographic applications. In *Proceedings of the 5th Innovations in Theoretical Computer Science Conference*, ITCS '14, pages 169–182, 2014. URL: https://dl.acm.org/doi/pdf/10.1145/2554797.2554815, doi:10.1145/2554797.2554815.

34 Liam Eagen and Ariel Gabizon. Protogalaxy: Efficient protostar-style folding of multiple instances. ePrint Report 2023/1106, 2023. URL: https://eprint.iacr.org/2023/1106.

35 Ariel Gabizon and Zachary J. Williamson. The turbo-plonk program syntax for specifying snark programs, 2019. URL: https://docs.zkproof.org/pages/standards/accepted-workshop3/proposal-turbo_plonk.pdf.

36 Ariel Gabizon and Zachary J. Williamson. plookup: A simplified polynomial protocol for lookup tables. ePrint Report 2020/315, 2020. URL: https://eprint.iacr.org/2020/315.

37 Alexander Golovnev, Jonathan Lee, Srinath T. V. Setty, Justin Thaler, and Riad S. Wahby. Brakedown: Linear-time and field-agnostic snarks for R1CS. In *Proceedings of the 43rd Annual International Cryptology Conference*, CRYPTO '23, pages 193–226, 2023. URL: https://eprint.iacr.org/2021/1043, doi:10.1007/978-3-031-38545-2_7.

38 Assimakis Kattis and Joseph Bonneau. Proof of necessary work: Succinct state verification with fairness guarantees. ePrint Report 2020/190, 2020. URL: https://eprint.iacr.org/2020/190.

39 Abhiram Kothapalli and Srinath Setty. Cyclefold: Folding-scheme-based recursive arguments over a cycle of elliptic curves. ePrint Report 2023/1192, August 2023. URL: https://eprint.iacr.org/2023/1192.

40 Abhiram Kothapalli and Srinath T. V. Setty. Hypernova: Recursive arguments for customizable constraint systems. In *Proceedings of the 44th Annual International Cryptology Conference*, CRYPTO '24, pages 345–379, 2024. URL: https://eprint.iacr.org/2023/573, doi:10.1007/978-3-031-68403-6_11.

41 Abhiram Kothapalli, Srinath T. V. Setty, and Ioanna Tzialla. Nova: Recursive zero-knowledge arguments from folding schemes. In *Proceedings of the 42nd Annual International Cryptology Conference*, CRYPTO '22, pages 359–388, 2022. URL: https://eprint.iacr.org/2021/370, doi:10.1007/978-3-031-15985-5_13.

42 Assa Naveh and Eran Tromer. PhotoProof: Cryptographic image authentication for any set of permissible transformations. In *Proceedings of the 37th IEEE Symposium on Security and Privacy*, S&P '16, pages 255–271, 2016. URL: https://ieeexplore.ieee.org/document/7546506, doi:10.1109/SP.2016.23.

43 Wilson D. Nguyen, Dan Boneh, and Srinath T. V. Setty. Revisiting the nova proof system on a cycle of curves. In *Proceedings of the 5th Conference on Advances in Financial Technologies*, AFT '23, pages 18:1–18:22, 2023. URL: https://eprint.iacr.org/2023/969, doi:10.4230/LIPICS.AFT.2023.18.

44 O(1) Labs. Mina Cryptocurrency, 2020. minaprotocol.org.

45 Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Proceedings of the 11th Annual International Cryptology Conference*, CRYPTO '91, pages 129–140, 1992. URL: https://link.springer.com/content/pdf/10.1007/3-540-46766-1_9.pdf.

46 Polygon Zero Team. Plonky2: Fast recursive arguments with plonk and fri. URL: https://github.com/0xPolygonZero/plonky2/blob/main/plonky2/plonky2.pdf.

47 Irving S. Reed, Gustave Solomon, and Kim Hamilton March. Polynomial codes over certain finite fields. *Journal of The Society for Industrial and Applied Mathematics*, 8:300–304, 1960.

48 Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. *SIAM J. Comp.*, 50(3), 2021. doi:10.1137/16M1096773.

49 Guy N. Rothblum, Salil P. Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In *Proceedings of the 45th ACM Symposium on the Theory of Computing*, STOC '13, pages 793–802, 2013. URL: https://privacytools.seas.harvard.edu/files/privacytools/files/stoc283fp-rothblum.pdf. doi:10.1145/2488608.2488709.

50 Daniel A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Trans. on Inf. Theory*, 42(6):1723–1731, 1996. URL: http://cs.yale.edu/homes/spielman/PAPERS/linearTimeIT.pdf, doi:10.1109/18.556668.

51 StarkWare. ethstark documentation. ePrint Report 2021/582, 2021. URL: https://eprint.iacr.org/2021/582.

52 Paul Valiant. Incrementally verifiable computation or proofs of knowledge imply time/space efficiency. In *Proceedings of the 5th Theory of Cryptography Conference*, TCC '08, pages 1–18, 2008. URL: https://iacr.org/archive/tcc2008/49480001/49480001.pdf, doi:10.1007/978-3-540-78524-8_1.

Sampling List Packings

Evan Camrud
Department of Mathematics and Statistics, Middlebury College, VT, USA

Ewan Davies[1]
Department of Computer Science, Colorado State University, Fort Collins, CO, USA

Alex Karduna
Department of Computer Science, Colorado State University, Fort Collins, CO, USA

Holden Lee
Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD, USA

Abstract

We initiate the study of approximately counting the number of list packings of a graph. The analogous problem for usual vertex coloring and list coloring has attracted substantial attention. For list packing the setup is similar, but we seek a full decomposition of the lists of colors into pairwise-disjoint proper list colorings. The existence of a list packing implies the existence of a list coloring, but the converse is false. Recent works on list packing have focused on existence or extremal results of on the number of list packings, but here we turn to the algorithmic aspects of counting and sampling.

In graphs of maximum degree Δ and when the number of colors is at least $\Omega(\Delta^2)$, we give a fully polynomial-time randomized approximation scheme (FPRAS) based on rapid mixing of a natural Markov chain (the Glauber dynamics) which we analyze with the path coupling technique. Some motivation for our work is the investigation of an atypical spin system, one where the number of spins for each vertex is much larger than the graph degree.

2012 ACM Subject Classification Theory of computation → Random walks and Markov chains; Mathematics of computing → Approximation algorithms

Keywords and phrases List packing, Graph colouring, Markov chains, Path coupling

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.24

Related Version *Full Version*: https://arxiv.org/abs/2402.03520

Funding *Ewan Davies*: supported in part by NSF grant CCF-2309707.

Acknowledgements We thank Colin McSwiggen and Semon Rezchikov for organizing Random Theory 2023, where this collaboration began. We thank Charlie Carlson and Guillem Perarnau for comments that led to improvements to the exposition of the paper.

1 Introduction

The classic graph coloring problem is to determine, for a graph $G = (V, E)$ and number of colors q, whether there is a coloring $f : V \to [q]$ such that f is proper in the sense that $f(u) \neq f(v)$ for every edge $uv \in E$. List coloring emerged in the late 20th century as an adversarial version of this problem [27, 13]. To define list coloring, we take a list size $q \in \mathbb{N}$ and assign to each $u \in V$ a list of allowed colors $L(u)$ of size q. One can think of the list assignment L as supplied by an adversary who might try to prevent the existence of a coloring respecting the lists. If, under any choices of such an assignment of lists, the graph G admits a proper coloring f such that $f(u) \in L(u)$ for every vertex u, then we say that G is q-list

[1] Corresponding author

© Evan Camrud, Ewan Davies, Alex Karduna, and Holden Lee;
licensed under Creative Commons License CC-BY 4.0

colorable (also known as q-choosable in some works). In some ways list coloring and classical graph coloring are similar, e.g., the complete graph on n vertices requires $q \geq n$ for both problems. But all bipartite graphs can be colored with two colors, while $K_{n,n}$ requires lists of length $\Omega(\log n)$ in the list coloring setting [13].

List coloring also arises naturally in various other ways. Notably, if we take a classic graph coloring problem and pre-color some vertices then we can express the problem of completing the coloring through list coloring. We set $L(u)$ to be the subset of the colors not used by any pre-colored vertex in the neighborhood of u (though these lists may not have equal sizes). The idea of list coloring has been extended in several interesting directions, and in this work we study a very recent and structured variant known as list packing [6]. The setup is the same as for list coloring with a fixed list size q, but instead of seeking just one proper coloring where each vertex is colored from its list, we seek q pairwise-disjoint proper colorings from the lists. We consider two colorings f and f' of a graph disjoint if, for all vertices u we have $f(u) \neq f'(u)$. We call this collection of q pairwise-disjoint proper list colorings a *list packing*. Given a graph G and list size q, if a list packing can be found for any lists of size q chosen by an adversary then we say that G is q-list packable.

One motivation for list packing in [6] is to challenge the state-of-the-art in list coloring. For example, a folklore result states that a bipartite graph of maximum degree Δ is q-list colorable for $q \geq (1 + o(1))\Delta/\log \Delta$ (see also [1] and a recent improvement of the leading constant due to Bradshaw [4]). Amongst other foundational results on list packing, the folklore result was matched for the significantly more structured notion of list packing in [6]. A notable difference between known results for list packing and list coloring is that a general graph of maximum degree Δ is q-list packable for $q \geq 2\Delta$ (improved to $q \geq 2\Delta - 2$ for $\Delta \geq 4$ in [8]); whilst such G are q-list colorable for $q \geq \Delta + 1$ (and even $q = \Delta$ if $\Delta \geq 3$ and G does not contain a clique on $\Delta + 1$ vertices [13]). Despite the gap in known results, there is scant evidence that more than $\Delta + 1$ colors are ever required for list packing [8, 6].

Early work on list packing has focused on the problem of existence, though many arguments for existence also provide efficient constructions of list packings. An extremal perspective on counting list packings was recently investigated by Kaul and Mudrock [21], and related problems where one seeks many list colorings or "flexible" list colorings are studied in [20, 7]. In this work we turn to the study of approximately counting the number of L-packings of a graph G with a fixed q-list assignment L. This is a natural question by analogy with the same questions for graph coloring and list coloring: is there an efficient procedure that, given a graph G and a number of colors q, approximates the number of proper q-colorings of G (e.g., to within a factor 2)? This question is well-studied in the field of approximate counting and sampling, and is an important test-bed for algorithmic techniques. A longstanding open question is the existence of such an approximate counting algorithm that works for all $q \geq \Delta + 1$ on graphs of maximum degree Δ. An influential collection of results using various techniques requires conditions such as $q \geq e\Delta + 1$ [2], $q \geq 2\Delta$ [17, 22], $q \geq 11\Delta/6$ [26], and even $q \geq (11/6 - \epsilon)\Delta$ for some small $\epsilon > 0$ [10, 9]. Another branch of research on the algorithmic aspects of counting graph colorings seeks to sample perfectly uniformly from the set of proper q-colorings of a graph. The pioneering result is due to Huber [15], with improvements and new techniques supplied in a number of later works [23, 22, 3, 16]. Interestingly, one application of such perfect samplers is to design approximate counting algorithms that can be faster than analogous approaches which use approximate samplers.

Motivated by simple and powerful ideas such as the Markov chain Monte Carlo approach to counting colorings due to Jerrum [17] (see also [25]), we seek similar results for list packing. The list packing problem, however, presents novel difficulties. Observe that finding a list

coloring gets strictly easier as the list size q grows. Supposing that one has a technique that works with lists of size q_0, then given larger lists one can take arbitrary subsets of the lists of size q_0 and apply the technique in a black-box fashion. In contrast, for list packing with larger q we are required to find ever more list colorings which must also be pairwise-disjoint. Given lists of size $q > q_0$, it is not at all clear how to extend an arbitrary collection of q_0 pairwise-disjoint list colorings to a full list packing. The methods of [8, 6] provide evidence that showing the *existence* of a list packing does get easier with larger q, albeit for less straightforward and general reasons. Whether *counting* list packings should get easier as q grows is another matter, but we confirm this principle in the setting of approximate counting by giving an algorithm that approximately counts list packings which works for all q large enough in terms of the maximum degree.

Approximate counting of combinatorial objects such as list colorings, independent sets, and matchings corresponds to approximating the so-called partition function of a spin system from statistical physics. We are interested in the study of counting and sampling list packings as it presents an unusual type of spin system. Typically, the number of spins available for each vertex in a spin system is small. In the Ising model of magnetism vertices take a spin from $\{+, -\}$, and in the (antiferromagnetic) Potts model associated with proper q-colorings, the spins are the q colors. Parameter ranges frequently studied for this model on graphs of maximum degree Δ include the case when q is close to $\Delta + 1$. The natural spin system associated with q-list packings, however, has $q!$ spins for each vertex. Since a vertex u has a list of q colors which we must decompose into q choices, one for each list coloring in the packing, the spins for u naturally correspond to permutations of the list $L(u)$. One of our contributions is to study a somewhat natural combinatorial problem which involves a spin system on bounded degree graphs with many more spins than commonly-studied examples. We hope that further insights into algorithmic techniques for approximate counting can be gained by studying an unusual spin system.

2 Results

Our main result is the existence of an approximate counting algorithm for list packings. We define an ϵ-approximation of a real number x as a real number y satisfying $e^{-\epsilon} \leq x/y \leq e^{\epsilon}$. We use the standard notion of a *fully polynomial-time randomized approximation scheme* (FPRAS) for a counting problem. This is a randomized algorithm that, given $\epsilon > 0$, yields an ϵ-approximation of the true answer in time polynomial in the input size and $1/\epsilon$.

Before we state the result, we need some more notation for list packings focusing on a specific list assignment L. Given a graph G we call an assignment $L : V(G) \to 2^{\mathbb{N}}$ of lists of colors to the vertices of G such that $|L(u)| = q$ for all vertices u a q-list assignment of G. A proper coloring f of G such that $f(u) \in L(u)$ for all vertices u is called an L-coloring, or a list coloring if the lists L are understood from context. Given a graph G and a q-list assignment L, we call a collection of q pairwise-disjoint L-colorings of G an L-packing. Note that an L-packing corresponds to permutation of the lists $L(v)$ for each vertex v.

▶ **Theorem 1.** *There is an absolute constant C such that the following holds. For any $\Delta \geq 1$ and $q \geq C\Delta^2$, let G be a graph of maximum degree Δ. Then for any q-list assignment L of G, there is an FPRAS for the number of L-packings of G.*

We prove this result by analyzing the so-called "heat-bath Glauber dynamics" for list packing. Briefly, the state space is the set of list packings of the graph and transitions involve resampling a uniform random valid permutation of the list of a uniform random choice of

vertex. We define "valid" carefully and analyze this Markov chain later, noting here that we prove Theorem 1 by establishing rapid mixing of this chain and hence an approximate sampling algorithm for a uniform L-packing under the same conditions.

A natural combinatorial problem on matchings in balanced bipartite graphs of large minimum degree emerges during the proof of Theorem 1, leading to a probabilistic result stated below that may be of independent interest. This is because the valid permutations of the list $L(v)$ correspond precisely to the perfect matchings in an auxiliary bipartite graph that we construct from the list packing on the neighbors of v. Throughout, we assume that Δ and q are fixed constants and do not analyze the case where they are allowed to depend on the number n of vertices of G as then algorithmic issues related to (perfectly) sampling perfect matchings in $2q$-vertex bipartite graphs become trivial[2]. We do not attempt to optimize the dependence of the running time on q or Δ.

Given a set A, we write $\mathcal{U}_A$ for the uniform distribution on A. When a fixed q is clear from context, let d_{C} be the Cayley metric on the symmetric group S_q. That is, $d_{\mathrm{C}}(\rho, \rho')$ is the minimum number of transpositions which one must compose with ρ to turn it into ρ'. This is merely graph distance on the Cayley graph of S_q generated by the transpositions. We associate perfect matchings in a balanced bipartite graph $H = ([q] \sqcup [q], E)$ with permutations $\rho \in S_q$ where $(i, \rho(i)) \in E$ for each i.

Given two random variables X, Y defined on the discrete probability spaces (Ω_X, p_X) and (Ω_Y, p_Y), a coupling of X, Y is a random variable $\gamma = (X', Y')$ defined on a probability space $(\Omega_X \times \Omega_Y, p_\gamma)$ such that X and X' have identical distributions and Y and Y' have identical distributions.

Given a Markov chain Z_t with state space Ω and transition matrix P, a coupling of Z_t is a Markov chain (X_t, Y_t) with state space $\Omega \times \Omega$ and transition matrix $\widehat{P}$ satisfying

$$\sum_{y' \in \Omega} \widehat{P}((x, y), (x', y')) = P(x, x'),$$
$$\sum_{x' \in \Omega} \widehat{P}((x, y), (x', y')) = P(y, y').$$

That is, each coordinate of the coupling is a faithful copy of Z_t, though the transitions of the coordinates are not necessarily independent.

▶ **Lemma 2.** *There are constants C_1, C_2 such that the following hold. Suppose that $q \geq C_1\Delta$, and let $H = (V, E)$ be a bipartite graph in which $|V| = 2q$ and the bipartition is balanced. Suppose that H has minimum degree at least $q - \Delta$. Let $e \in E$ be an edge of H, let $\mathcal{L}$ be the set of perfect matchings of H containing e, and let $\mathcal{R}$ the set of perfect matchings of H not containing e. Then there is a coupling γ of $\mathcal{U}_{\mathcal{R}}$ and $\mathcal{U}_{\mathcal{L} \cup \mathcal{R}}$ such that*

$$\mathbb{E}_{(\rho, \rho') \sim \gamma}[d_{\mathrm{C}}(\rho, \rho')] \leq \frac{C_2\Delta}{q}.$$

This lemma is the main bottleneck for improving the dependence of q on Δ in our main theorem. In particular, removing the factor of Δ would also improve Theorem 1 by a factor of Δ.

[2] Given a bipartite graph of constant order which contains a perfect matching, sampling a perfect matching uniformly at random can be done in constant time, e.g. by exhaustively enumerating the perfect matchings and selecting one at random.

3 Technical overview

We prove Theorem 1 by the standard Markov chain Monte Carlo approach and the path coupling technique of Bubley and Dyer [5]. We study an ergodic Markov chain – the heat-bath Glauber dynamics – whose stationary distribution is uniform on the list packings of a graph, and use path coupling to show rapid mixing. Then a well-known and generic reduction from counting to sampling yields Theorem 1. Path coupling reduces the potentially challenging task of proving rapid mixing of a Markov chain to designing a coupling on adjacent states according to some graph Γ on the state space Ω. We largely follow the notation of [12] and consider

$$\mathcal{M} := \{P^t p_0\}_{t=0}^{\infty},$$

an ergodic Markov chain on state space Ω with initial distribution p_0 and transition operator P. We denote the (unique) stationary distribution by π, and denote by $p_t := P^t p_0$ the distribution of the state of the chain $\mathcal{M}$ after t steps. We use the standard notion of mixing time of Markov chains given by

$$t_{\text{mix}}(\epsilon) := \max_{p_0} \min \{t \geq 0 : d_{\text{TV}}(p_t, \pi) \leq \epsilon\},$$

where d_{TV} is total variation distance.

Typically, we write ω and ω' for states of the chain before a transition and $\sigma := P\omega$, $\sigma' := P\omega'$ for the (random) states after one step of the chain from ω and ω' respectively.

▶ **Theorem 3** (Bubley and Dyer [5], see also [12]). *Let Ω be the state space of a Markov chain $\mathcal{M}$ and let Γ be a weighted, directed graph on vertex set Ω with edge weights in $\mathbb{N}$. Let δ be the quasi-metric[3] on Ω given by taking shortest paths in Γ, and suppose that $\delta(\omega, \omega') \leq D < \infty$ for all $\omega, \omega' \in \Omega$.*

For each $(\omega, \omega') \in E(\Gamma)$, suppose that we have a coupling γ, of the random variables σ (with distribution $P\omega$) and σ' (with distribution $P\omega'$) for which $\mathbb{E}_{(\sigma,\sigma')\sim\gamma}[\delta(\sigma, \sigma')] \leq \beta\delta(\omega, \omega')$. Then if $\beta < 1$, we have $t_{\text{mix}}(\epsilon) \leq \log(D/\epsilon)/(1-\beta)$.

The theorem may seem rather abstract, so we briefly discuss a well-known application to list coloring as a warm-up to the main argument for list packing. Let G be a graph of maximum degree Δ and let L be a q-list assignment of G. Let Ω be the set of L-colorings of G, and let $\mathcal{M}$ be the heat-bath Glauber dynamics on Ω, defined as follows. Note that states $\omega \in \Omega$ are colorings and hence functions $V(G) \to \mathbb{N}$. A transition of $\mathcal{M}$ from a state ω is performed by choosing a vertex $u \in V(G)$ uniformly at random, sampling a color c in $L(u) \setminus \omega(N(u))$, and moving to the state σ with $\sigma(u) = c$ and $\sigma(v) = \omega(v)$ for all $v \neq u$. That is, the transition operator P is defined via

1. sampling a vertex $u \sim \mathcal{U}_{V(G)}$,
2. sampling a color $c \sim \mathcal{U}_{L(u)\setminus\omega(N(u))}$,
3. defining σ by $\sigma(v) := \begin{cases} c & v = u \\ \omega(v) & v \neq u \end{cases}$, and moving to the state σ.

Note that $L(u) \setminus \omega(N(u))$ is the set of available colors for u. These colors are in the list $L(u)$ but are not used by the coloring ω on the neighbors of u so setting $\omega(u)$ to one of them yields a valid list coloring. This chain is reversible, and when $q \geq \Delta + 2$ it is ergodic [18, Exercises 4.1] with stationary distribution uniform on Ω.

[3] that is, a function which satisfies the conditions of a metric except symmetry

For the purpose of constructing the couplings required by Theorem 3, let Γ be the weighted, directed graph on Ω where (ω, ω') is an edge if and only if the colorings ω, ω' differ at exactly one vertex and let all edge weights be 1. From the proof of ergodicity [18] it follows that we can take $D = (\Delta + 1)n$ in Theorem 3 and define couplings as follows. Let $(\omega, \omega') \in E(\Gamma)$ and suppose that the colorings ω and ω' differ at the vertex v^*. Sample $u \in V(G)$ uniformly at random and update the color of u in both chains (intuitively, this decision helps the chains coalesce). That is, the distribution γ of (σ, σ') is defined by the Markov transition $(\omega, \omega') \overset{P_\gamma}{\mapsto} (\sigma, \sigma')$ itself defined by

1. sampling a vertex $u \sim \mathcal{U}_{V(G)}$,
2. sampling a pair of available colors (a, b) from a distribution γ_c such that

$$\gamma_c := \underset{\substack{\gamma' \text{ is a coupling of} \\ \mathcal{U}_{L(u)\setminus\omega(N(u))} \text{ and } \mathcal{U}_{L(u)\setminus\omega'(N(u))}}}{\arg\max} \Pr_{(a,b)\sim\gamma'}(a = b) \tag{1}$$

3. defining σ and σ' by updating the color of u in each to a and b respectively:

$$\sigma(v) := \begin{cases} a & v = u \\ \omega(v) & v \neq u \end{cases}, \qquad \sigma'(v) := \begin{cases} b & v = u \\ \omega(v) & v \neq u \end{cases},$$

and moving to the state (σ, σ').

Observe that γ_c is defined as the coupling on the uniform distributions of available colors, $\mathcal{U}_{L(u)\setminus\omega(N(u))}$ and $\mathcal{U}_{L(u)\setminus\omega'(N(u))}$, which maximizes the probability the colors are the same. The important property of this definition in terms of applying Theorem 3 is that this coupling minimizes the expectation of the discrete metric on the sample.

If $u \in N(v^*)$ the sets of available colors for u can be different making γ_c nontrivial. In this case, let $C = \omega(N(u) \setminus \{v^*\})$ and note that the two sets of available colors are $A = (L(u) \setminus C) \setminus \{\omega(v^*)\}$ and $B = (L(u) \setminus C) \setminus \{\omega'(v^*)\}$. We wish to couple $\mathcal{U}_A$ and $\mathcal{U}_B$ such that the probability of choosing the same color is maximized, and the best general coupling is not too hard to find.

▶ **Lemma 4** (See e.g., [18, Lemma 4.10]). *Let U be a finite set and $A, B \subset U$. Then there is a coupling γ_c of $\mathcal{U}_A$ and $\mathcal{U}_B$ such that*

$$\Pr_{(a,b)\sim\gamma_c}(a = b) = \frac{|A \cap B|}{\max\{|A|, |B|\}},$$

where (a, b) is a random element of $A \times B$ chosen according to the coupling γ_c.

Let $m = \max\{|A|, |B|\}$. We have $A \cap B = (L(u) \setminus C) \setminus \{\omega(v^*), \omega'(v^*)\}$. Checking the four cases according to whether each of $\omega(v^*)$ and $\omega'(v^*)$ are in $L(u) \setminus C$, we observe that $|A \cap B| \geq m - 1$. By Lemma 4, we can ensure that the two chains choose the same color with probability at least $1 - 1/m$. Considering the definitions of A and B, we also have $m \geq q - \Delta$. Returning to the analysis of the coupling γ described above, we have

$$\mathbb{E}_{(\sigma,\sigma')\sim\gamma}[\delta(\sigma, \sigma')] \leq 1 + \frac{1}{n}\left(-1 + \frac{\Delta}{q - \Delta}\right).$$

This comes from the facts that $\delta(\omega, \omega') = 1$ by assumption, the probability $1/n$ that we successfully reduce the distance by 1 in an update to v^*, and the probability of at most Δ/n that we choose to update a neighbor of v^*, and in this case fail to choose the same color in the coupling of the color choice in each chain given by Lemma 4 (which occurs with probability at most $1/(q - \Delta)$ given that we update a neighbor of v^*). Set β to the right-hand side above and solve for $\beta < 1$ to obtain $q > 2\Delta$.

Well-known works that first studied this technique [17, 5, 18] give a similar proof, though Jerrum [17] manually constructed a coupling of the Markov chain and did not appeal to path coupling. With path coupling, it is slightly easier to study a variant of the Markov chain known as the "Metropolis Glauber dynamics" where the transition is defined slightly differently, though the same lower bound on q is required in the argument for this chain.

Our argument for list packing follows the same outline as above using the heat-bath dynamics for list packings, but it is much harder to construct the coupling. In particular, we need an analogue of Lemma 4 for the much more intricate combinatorial setting of list packings. We describe this in the next subsection.

3.1 Coupling perfect matchings

In the list packing setting, each spin is a permutation of a list. The central problem one faces when adapting the above sketch to list packing is the issue of coupling the choice of available permutations in the case that we are updating the spin of u in two copies of the Glauber dynamics which differ at a neighbor $v \in N(u)$. It turns out (see e.g. [6] and earlier works such as [24]) that there is an auxiliary bipartite graph in which available permutations for u correspond to perfect matchings.

Given a graph G with q-list assignment L, let Ω be the set of L-packings of G, and let $\omega \in \Omega$. Let $f_1, \dots, f_q$ be the L-colorings represented by ω. To be explicit, suppose that for each vertex v of G we write $L(v)$ in ascending order as $c_{1,v}, \dots, c_{q,v}$. Then we identify $V(G)$ with $[n]$ and consider Ω as a subset of S_q^n such that with $v \in [n]$ we define $f_1, \dots, f_q : V(G) \to \mathbb{N}$ by $f_i(v) = c_{\omega_v(i),v}$, where we interpret ω_v as a permutation of $L(v)$ so that $\omega_v(i)$ is the color assigned to v in the ith coloring in the packing. This notation has the advantage that many of the dependencies are explicit, but to avoid a proliferation of subscripts we omit them where it is possible to fix some context in advance.

For a vertex $u \in V(G)$, we construct the *availability graph* $H_u = H_u(G, L, \omega)$ as follows. We consider the vertex set of H_u as $[q] \sqcup [q]$, the disjoint union of two copies of the set $[q]$. The left copy consists of "packing indices" and the right copy consists of "list indices". For clarity, the edges of H_u are considered oriented from left to right so that (i, j) joins packing index i to list index j. Suppose that $L(u) = \{c_1, \dots, c_q\}$ is supplied in some fixed order. Then in H_u we include each edge $(i, j) \in [q]^2$ such that c_j is an available color for u in the coloring f_i. That is, (i, j) is an edge of H_u if and only if $c_j \notin f_i(N(u))$. One can check the defintions and observe that perfect matchings in H_u correspond to the available permutations $\rho \in S_q$ for u in the sense that setting ω_u to an available ρ yields a valid list packing. The heat-bath Glauber dynamics for L-packings thus works in much the same way as for L-colorings. The transition from a state ω is defined by choosing a vertex $u \in V(G)$ uniformly at random, and then a perfect matching in $H_u(G, L, \omega)$ uniformly at random. It is straightforward to check that the chain is reversible and has uniform stationary distribution; we prove that it is ergodic (for large enough q) in Lemma 7.

An important consideration when G has maximum degree Δ is that H_u has minimum degree $q - \Delta$. This follows from the properties of a list packing: at packing index i any color that is not available must be used by f_i on a neighbor of u and there are at most Δ such neighbors. Similarly, for a color c_j with color index j, any packing index i in which c_j is not available is explained by c_j being used by f_i on $N(u)$. Since the colorings in a packing are pairwise-disjoint, each such index must be due to distinct neighbors of u, of which there are at most Δ. It is useful to observe that for any $\omega \in S_q^n$, even one that may not be a proper list packing in the sense that the q list colorings it represents may not be proper, the definition of availability graph still makes sense and the observation on the minimum

degree still applies. That is, the key property of q-list packings on graphs of maximum degree Δ that yields the minimum degree bound $q - \Delta$ is that the list colorings represented are pairwise-disjoint. This fact is convenient in the proof of Lemma 5.

To construct the coupling we consider the weighted, directed graph Γ on Ω where (ω, ω') is an edge if and only if the list packings ω and ω' differ at exactly one vertex. For the edge $(\omega, \omega') \in E(\Gamma)$, let v be the unique vertex at which the two packings differ, and assign weight $d_{\mathrm{C}}(\omega_v, \omega'_v)$ to the edge. As with list coloring, the key computation is how much the expected distance changes for one step of the coupling in the case that we start at (ω, ω') and update a neighbor u of the unique vertex v at which ω and ω' differ. We prove the following result which plays the role of Lemma 4 in our proof.

▶ **Lemma 5.** *There is a universal constant C such that if $q > C\Delta^2$ the following holds.*

Let $(\omega, \omega') \in E(\Gamma)$ be an edge of weight ψ in Γ and let v be the unique vertex at which ω and ω' differ. Let $u \in N(v)$ and consider the availability graphs H and H' for u in the packings ω and ω' respectively.

Then there exists a coupling γ of the uniform distributions on perfect matchings on H and H' which satisfies

$$\mathbb{E}_{(\rho,\rho')\sim\gamma}[d_{\mathrm{C}}(\rho, \rho')] \leq \frac{\psi}{2\Delta}.$$

Note that in the conclusion any bound strictly better than ψ/Δ suffices for an application of Theorem 3, and one might expect to obtain results in the case of exactly ψ/Δ as well [17, 5, 12]. We do not attempt to optimize the constant C in our argument and hence $\psi/(2\Delta)$ is sufficient. Lemma 5 is a simple corollary of Lemma 2, and with these results in hand the rest of the argument for Theorem 1 is standard.

3.2 Organization

In Section 4 we define a Markov chain on list packings, prove ergodicity and establish rapid mixing given our results on coupling perfect matchings. In Section 5 we prove Lemmas 2 and 5. We conclude with some remarks in Section 6.

4 Glauber dynamics for list packing

In this section we fix an n-vertex graph G of maximum degree Δ, a q-list assignment L of G, and let Ω be the set of L-packings of G.

We consider heat-bath Glauber dynamics $\mathcal{M} = \mathcal{M}(G, L)$ for list packing. Given a state $\omega \in \Omega$ and a vertex $u \in V(G)$, we say that a permutation ρ in S_q is *available* for u in ω if setting ω_u to ρ yields a valid list packing. Transitions of $\mathcal{M}$ are defined as follows. From state $\omega \in \Omega$, choose vertex $u \in V(G)$ uniformly at random, choose an available permutation $\rho \in S_q$ uniformly at random, and let the new state be σ given by $\sigma_u = \rho$ and $\sigma_v = \omega_v$ for $v \neq u$. It is straightforward to check that this is a reversible Markov chain on Ω with uniform stationary distribution. The question of ergodicity is less straightforward, though the standard argument for list coloring adapts easily. We require a simple corollary of Hall's classic result on perfect matchings in bipartite graphs.

▶ **Lemma 6** (Corollary of Hall's theorem [14])**.** *Let H be a bipartite graph with q vertices on each side and minimum degree $d \geq q/2$. Then H contains a perfect matching.*

Sketch proof. Let the bipartition be $A \sqcup B$ and consider a subset $X \subset A$. If $1 \leq |X| \leq d$ then $|N(X)| \geq d \geq |X|$. If instead $d+1 \leq |X| \leq q$ then $N(X)$ must be all of B since for each neighborhood of a vertex in B must intersect X. The result follows from Hall's theorem. ◀

▶ **Lemma 7.** *For $q \geq 2\Delta + 2$ the Markov chain $\mathcal{M}$ is ergodic.*

Proof. Because there are self-transitions at every state, it suffices to show that the (finite) state space is connected.

Every state $\omega \in \Omega$ can be connected to an arbitrary $\omega' \in \Omega$ as follows. Label the vertices of G with the integers $1, \ldots, n$ arbitrarily, and for $i = 1, \ldots, n$, sequentially turn ω_i into ω'_i as follows. If setting ω_i equal to ω'_i is not possible (i.e. ω'_i is not available for i in ω) then it's because some neighbor j of i with $j > i$ uses a color at a particular packing index which conflicts with ω_i. To solution is to repack each such neighbor j in ω, i.e. change ω_j to a new permutation of $L(j)$, in turn such that the repacking is proper *and* avoids any such conflicts. These repacking steps are steps of the chain.

To perform the repacking, we find a perfect matching in a suitable modification of the availability graph $H'_j = H_j(G, L, \omega')$. Recall that perfect matchings in H'_j correspond to the available permutations ρ for u in ω'. We have an additional condition on the permutation ρ that we seek, namely that ρ does not correspond to a color choice for j that is incompatible with turning ω_i into ω'_i. We can encode this in the availability graph by deleting an edge (a, b) such that the b-th color in $L(j)$ is used at packing index a in color choices represented by ω'_i. After this modification, the availability graph has minimum degree $q - \Delta - 1$, so the condition $q \geq 2\Delta + 2$ allows for an application of Lemma 6. This shows that the necessary repackings exist, and thus that Ω is connected by transitions of the chain that occur with positive probability. ◀

▶ **Theorem 8.** *The mixing time $t_{\text{mix}}(\epsilon)$ of $\mathcal{M}$ is at most $O(n \log(n/\epsilon))$.*

Proof. We apply Theorem 3 with the following coupling defined on edges of the weighted graph Γ on Ω such that (ω, ω') is an edge of weight $d_{\text{C}}(\omega_v, \omega'_v)$ whenever ω and ω' differ at a single vertex v. Lemma 7 shows that Γ is connected, and we note that the diameter D is at most $(\Delta + 1)(q - 1)n = O(n)$. This is because the sequence of steps constructed in Lemma 7 consists of edges of Γ, the total number of steps is at most $(\Delta + 1)n$ because we repack each vertex v at most once for each neighbor of v to avoid conflict and at most once more to agree with ω'. As the Cayley distance on S_q takes values in $\{0, 1, \ldots, q - 1\}$, the diameter bound follows.

Let (ω, ω') be an edge of Γ of weight ψ. We define the coupling as follows. We choose $u \in V(G)$ uniformly at random and update u in both packings. If $u \notin N(v)$ then the sets of available permutations of $L(u)$ in both packings are identical and we choose one uniformly at random to use in both chains. If $u = v$ this results in a distance of zero, else the distance is unchanged. If $u \in N(v)$ then we use the coupling of Lemma 5 to choose the permutations of $L(u)$ in the packings. Let (σ, σ') be the random state after one step of the coupling started from (ω, ω'). Then

$$\mathbb{E}[\delta(\sigma, \sigma')] \leq \psi + \frac{1}{n}\left(-\psi + \Delta\frac{\psi}{2\Delta}\right) = \psi\left(1 - \frac{1}{2n}\right).$$

Theorem 3 now gives mixing time $t_{\text{mix}}(\epsilon) = O(n \log(n/\epsilon))$. ◀

The proof of Theorem 1 using Theorem 8 is now entirely standard. We sketch the argument here. Counting list packings is a *self-reducible* problem in the sense of [19] and so having an almost-uniform sampler, which follows from running the Markov chain for

polynomially many steps, is equivalent to having an FPRAS. Concretely, construct a sequence $G = G_m \supset \cdots \supset G_1 \supset G_0 = (V(G), \emptyset)$ of graphs by starting from G and removing an arbitrary edge to form the next member of the sequence. For a fixed q-list assignment L of G we can let Ω_i be the set of L-packings of G_i and write

$$|\Omega_m| = |\Omega_0| \prod_{i=1}^{m} \frac{|\Omega_i|}{|\Omega_{i-1}|}.$$

We have $|\Omega_0| = (q!)^n$ and can estimate each ratio in the product as it's the probability that when we choose a unformly random $\omega \in \Omega_{i-1}$ we have $\omega \in \Omega_i$. We also note that $\frac{1}{1+q!} \leq \frac{|\Omega_i|}{|\Omega_{i-1}|} \leq 1$ because $\Omega_i \subset \Omega_{i-1}$ and we can construct the following bipartite graph B on $(\Omega_{i-1} \setminus \Omega_i) \sqcup \Omega_i$. Let uv be the edge removed from G_i to form G_{i-1}, and include the edge (ω, ω') in B if ω' can be obtained from Ω by permuting the list of u. Then any $\omega \in \Omega_{i-1} \setminus \Omega_i$ is connected to at least one element in Ω_i by Lemma 6, and there are at most $q!$ ways to permute $L(u)$ so from the other side the degrees are at most $q!$. We do not attempt to optimize this argument; more intricate arguments yield stronger lower bounds, but we are merely interested in a bound independent of n. This observation lets us repeat the analysis of Jerrum [17] for the case of colorings in the setting of list packings. Briefly, we use multiple copies of the almost uniform sampler offered by the Markov chain to estimate each ratio. To make this work, one has to bound the variance of the estimator for each ratio and manage the overall error with Chebyshev's inequality, but this is standard. See e.g., [17].

5 Coupling matchings

In this section we prove Lemmas 2 and 5. We first collect some results on auxiliary bipartite graphs.

▶ **Lemma 9.** *Let $G = (V, E)$ be a bipartite graph with bipartition $V = L \cup R$. Let μ_L and μ_R be probability distributions over L and R. Let*

$$p = \min_{A \subseteq L} [1 - \mu_L(A) + \mu_R(N(A))].$$

There exists a coupling γ of μ_L and μ_R such that when $(v, w) \sim \gamma$, with probability p, vw is an edge of G.

Proof. This follows from the max-flow min-cut theorem after introducing a source vertex s, a sink vertex t, and introducing the following edges:

1. for $v \in L$, the edge sv with capacity $\mu_L(v)$,
2. for $v \in R$, the edge vt with capacity $\mu_R(v)$,
3. for each edge $e \in G$, an edge with capacity ∞.

A max flow corresponds to a coupling γ of the type we require with maximum probability that vw is an edge. The value of the max flow is the same as the value of the min cut. If S, T is a finite cut with $s \in S$, $t \in T$, letting $A = S \cap L$, $B = S \cap R$, we must have $N(A) \subseteq B$, and the value is $\mu_L(A^c) + \mu_R(B) \geq \mu_L(A^c) + \mu_R(N(A))$. Equality is achieved for $B = N(A)$. Taking the minimum over A then gives the lemma. ◀

▶ **Corollary 10.** *Let $G = (V, E)$ be a bipartite graph with bipartition $V = L \cup R$. Suppose that each vertex in L has degree contained in $[m_L, M_L]$ and each vertex in R has degree contained in $[m_R, M_R]$. Then there exists a coupling γ of the uniform distributions $\mathcal{U}_L$ and $\mathcal{U}_R$ on L and R respectively such that when $(v, w) \sim \gamma$, with probability at least $\frac{m_L m_R}{M_L M_R}$, vw is an edge of G.*

Proof. For each subset $A \subseteq L$, we can bound the number of edges between A and $N(A)$:

$$m_L|A| \leq |E(A, N(A))| \leq M_R|N(A)|.$$

We also have

$$m_R|R| \leq |E| \leq M_L|L|.$$

Then for any set $A \subseteq L$,

$$\frac{|N(A)|/|R|}{|A|/|L|} = \frac{|N(A)|}{|A|} \cdot \frac{|L|}{|R|} \geq \frac{m_L m_R}{M_L M_R}.$$

Then for any such A,

$$1 - \mathcal{U}_L(A) + \mathcal{U}_R(N(A)) \geq 1 - \mathcal{U}_L(A) + \frac{m_L m_R}{M_L M_R} \cdot \mathcal{U}_L(A) \geq \frac{m_L m_R}{M_L M_R}.$$

The conclusion follows from Lemma 9. ◀

We are now ready to prove Lemma 2, which we restate for convenience.

▶ **Lemma 2.** *There are constants C_1, C_2 such that the following hold. Suppose that $q \geq C_1\Delta$, and let $H = (V, E)$ be a bipartite graph in which $|V| = 2q$ and the bipartition is balanced. Suppose that H has minimum degree at least $q - \Delta$. Let $e \in E$ be an edge of H, let $\mathcal{L}$ be the set of perfect matchings of H containing e, and let $\mathcal{R}$ the set of perfect matchings of H not containing e. Then there is a coupling γ of $\mathcal{U}_{\mathcal{R}}$ and $\mathcal{U}_{\mathcal{L}\cup\mathcal{R}}$ such that*

$$\mathbb{E}_{(\rho,\rho')\sim\gamma}[d_{\mathrm{C}}(\rho, \rho')] \leq \frac{C_2\Delta}{q}.$$

Proof. Label the bipartitions by $[q]$. Without loss of generality the edge e is $(1, 1)$. We define an auxiliary bipartite graph on $\mathcal{L} \cup \mathcal{R}$ as follows. For $\rho \in \mathcal{L}$ and $\rho' \in \mathcal{R}$, connect ρ and ρ' by an edge if $\rho(1\,i\,j) = \rho'$ for some $i, j \in [q] \setminus \{1\}$ with $i \neq j$, i.e., they differ by a 3-cycle containing 1. Note that here we use standard group-theoretic notation for permutations in S_q so that e.g. $\rho(1\,i\,j)$ is the product of the permutation ρ and the permutation $(1\,i\,j)$ (in standard cycle notation).

We bound the degrees of arbitrary matchings $\rho \in \mathcal{L}$ and $\rho' \in \mathcal{R}$. Given ρ, we count the number of i, j for which $\rho(1\,i\,j) \in \mathcal{R}$. Let $N(i) = \{j : (i, j) \in E\}$ and $N'(j) = \{i : (i, j) \in E\}$. Given $\rho \in L$, we know that $\rho(1) = 1$. We choose $j \in N'(1) \setminus \{1\}$; there are at least $q - \Delta - 1$ choices. Any possible value of i must be in the set $S_- \setminus \{j\}$, where $S_- := \{\rho^{-1}(k) : k \in N(1) \setminus \{1\}\}$; $S_- \setminus \{j\}$ has size at least $q - \Delta - 2$. A valid pair (i, j) is exactly one where $j \in N'(1) \setminus \{1\}$, $i \in S_- \setminus \{j\}$, and $(i, \rho(j)) \in E$. Since at most $\Delta + 1$ of edges $(i, \rho(j))$, $j \in N'(1) \setminus \{1\}$ can land outside E, at least $q - 2\Delta - 3$ of these edges are valid, i.e., there are at least $(q - \Delta - 1)(q - 2\Delta - 3)$ valid choices of (i, j). There are at most $(q - 1)(q - 2)$ choices.

Next, given $\rho' \in \mathcal{R}$, we count the number of $i, j \in [q] \setminus \{1\}$, $i \neq j$ for which $\rho' = \rho(1\,i\,j)$ where $\rho \in \mathcal{L}$. First, note we must have $\rho'(j) = 1$. The requirement on i is that $i \in N'(\rho'(1))$ and $\rho'(i) \in N(j)$. There are at least $q - \Delta - 2$ indices besides 1 and j satisfying each condition, so at least $q - 2\Delta - 4$ possible indices. There are at most $q - 2$ choices. By Corollary 10, there is a coupling γ of $\mathcal{U}_{\mathcal{L}}$ and $\mathcal{U}_{\mathcal{R}}$ such that with probability at least

$$p := \frac{(q - \Delta - 1)(q - 2\Delta - 3)}{(q - 1)(q - 2)} \cdot \frac{q - 2\Delta - 4}{q - 2},$$

$(\rho, \rho') \in E$ are connected by an edge and hence have Cayley distance at most two. For appropriate choices of $C_1, K > 0$, $1 - p \leq K\Delta/q$ for $q \geq C_1\Delta$. For this coupling γ, we hence have

$$\mathbb{E}_{(\rho,\rho')\sim\gamma}[d_{\mathrm{C}}(\rho, \rho')] \leq \frac{K\Delta}{q} \cdot q + \left(1 - \frac{K\Delta}{q}\right) \cdot 2 \leq K'\Delta$$

for an appropriate constant $K' > 0$.

Next, note that we can define a coupling between $\mathcal{U}_{\mathcal{L}\cup\mathcal{R}}$ and $\mathcal{U}_{\mathcal{R}}$ as a mixture of the identity coupling between $\mathcal{U}_{\mathcal{R}}$ and $\mathcal{U}_{\mathcal{R}}$ (with probability $\frac{|\mathcal{R}|}{|\mathcal{L}|+|\mathcal{R}|}$) and the above coupling (with probability $\frac{|\mathcal{L}|}{|\mathcal{L}|+|\mathcal{R}|}$). The expected distance for this coupling is then

$$\frac{|\mathcal{L}|}{|\mathcal{L}| + |\mathcal{R}|} \cdot K'\Delta \leq \frac{q-2}{(q-2) + (q-\Delta-1)(q-2\Delta-3)} \cdot K'\Delta \leq \frac{C_2\Delta}{q}$$

for appropriate $C_2 > 0$, as needed. ◀

▶ **Lemma 5.** *There is a universal constant C such that if $q > C\Delta^2$ the following holds.*

Let $(\omega, \omega') \in E(\Gamma)$ be an edge of weight ψ in Γ and let v be the unique vertex at which ω and ω' differ. Let $u \in N(v)$ and consider the availability graphs H and H' for u in the packings ω and ω' respectively.

Then there exists a coupling γ of the uniform distributions on perfect matchings on H and H' which satisfies

$$\mathbb{E}_{(\rho,\rho')\sim\gamma}[d_{\mathrm{C}}(\rho, \rho')] \leq \frac{\psi}{2\Delta}.$$

Proof. Formally, we proceed by induction on ψ, though we need a more general statement for the induction hypothesis. It is convenient to relax the requirement that ω and ω' are valid list packings. For the application, it is important that the pair (ω, ω') are valid list packings which agree on every vertex except u, but we construct the coupling of perfect matchings in H and H' using a sequence of *near-valid* list packings $\omega'' \in S_q^n$ in the sense that ω'' agrees with ω and ω' on all vertices except u, but we allow the (pairwise-disjoint) colorings it represents to have monochromatic edges incident to v. We can still construct availability graphs for u in these near-valid packings and consider their sets of perfect matchings for the purposes of constructing an eventual coupling of the perfect matchings in H and H'. With these definitions in place, the generalization that we prove by induction is the statement obtained by replacing the assumption that ω and ω' are valid packings with the assumption that they are near-valid.

The base case is $\psi = 0$ in which the trivial coupling suffices as H and H', and hence their sets of perfect matchings, are identical.

The induction step follows from Lemma 2. The fact that $d_{\mathrm{C}}(\omega_v, \omega'_v) = \psi$ means that there is a sequence of transpositions $\tau_1, \ldots, \tau_\psi$ such that $\omega_v = \tau_\psi \cdots \tau_1 \omega'_v$. Let H'' be the availability graph of the vertex u in the near-valid packing ω'' such that $\omega''_v = \tau_\psi \omega_v$ and ω'' agrees with ω on all other vertices. By induction, there is a coupling γ' of the uniform distributions on perfect matchings in H'' and H' such that

$$\mathbb{E}_{(\rho'',\rho')\sim\gamma'}[d_{\mathrm{C}}(\rho'', \rho')] \leq \frac{\psi - 1}{2\Delta}.$$

Without loss of generality, suppose that ω_v is the identity. Let $\tau_\psi = (i\, j)$, and note that this gives

$$E(H) \setminus E(H'') \subset \{(i,j),(j,i)\} \qquad \text{and} \qquad E(H'') \setminus E(H) \subset \{(i,i),(j,j)\}.$$

This is because any difference between the edges of H and H' is explained by applying τ to ω_v. When τ is a transposition, we swap the packing index of the coloring at which two colors in the list of v are used, which can swap two edges of the complement of the availability graph. It can be the case that the swapped color indices refer to different colors in $L(u)$ which is why we do not have equality.

We start with the case that

$$E(H) \setminus E(H'') = \{(i,j),(j,i)\} \qquad \text{and} \qquad E(H'') \setminus E(H) = \{(i,i),(j,j)\},$$

the other cases are similar. Let X be the set of perfect matchings in H and let X'' be the set of perfect matchings in H''. We seek a coupling of $\mathcal{U}_X$ and $\mathcal{U}_{X''}$ which we will combine with the coupling γ' to obtain the desired result.

We apply Lemma 2 to H and $H-(i,j)$, yielding a coupling γ_1 of $\mathcal{U}_X$ and $\mathcal{U}_Y$, where Y is the set of perfect matchings in $H-(i,j)$. We can apply Lemma 2 again to $H-(i,j)$ and $H-(i,j)-(j,i)$, yielding a coupling γ_2 of $\mathcal{U}_Y$ and $\mathcal{U}_Z$, where Z is the set of perfect matchings in $H-(i,j)-(j,i)$. There is a slight technicality here as the minimum degree of $H-(i,j)$ is $q-\Delta-1$, but this can be handled by setting Δ to $\Delta+1$ and adjusting the constants slightly. Analogously, starting from H'', we construct a coupling γ_1'' of $\mathcal{U}_{X''}$ and $\mathcal{U}_{Y''}$ and a coupling γ_2'' of $\mathcal{U}_{Y''}$ and $\mathcal{U}_Z$, where Y'' is the set of perfect matchings in $H''-(i,i)$. A careful composition of these couplings gives the result. The composition of these couplings yields a distribution on $X \times Y \times Z \times Y'' \times X''$ which is uniform on each individual set in the Cartesian product, and such that the expected distance between permutations from adjacent sets in the Cartesian product is at most $C_2\Delta/q$. Taking the first and last coordinate yields a coupling γ'' of $\mathcal{U}_X$ and $\mathcal{U}_{X''}$ such that

$$\mathbb{E}_{(\pi,\pi'')\sim\gamma''}[d_{\mathrm{C}}(\pi,\pi'')] < \frac{4C_2\Delta}{q}.$$

This can be combined with γ' obtained by induction in the same way. Simple composition yields a distribution on $X \times X'' \times X'$ which is uniform on each individual set in the Cartesian product. Taking the first and last coordinates we have a coupling γ of $\mathcal{U}_X$ and $\mathcal{U}_{X'}$ such that

$$\mathbb{E}_{(\pi,\pi')\sim\gamma}[d_{\mathrm{C}}(\pi,\pi')] < \frac{\psi-1}{2\Delta} + \frac{4C_2\Delta}{q}.$$

Since we assume $q \geq C\Delta^2$, for a large enough C this is at most $\psi/(2\Delta)$ as required. The other cases proceed similarly, but require fewer applications of Lemma 2 and yield a stronger upper bound. ◀

6 Concluding remarks

Many natural questions remain unanswered. We have chosen to extend some of the most fundamental techniques for counting list colorings to list packings, but there are many more recent improvements to consider. Dyer and Greenhill [11] study a Markov chain on (list) colorings whose transitions are defined by properly recoloring both endpoints of a uniform random edge and show that its mixing time is less than that of Glauber dynamics studied in [17, 25]. The flip dynamics employed by Vigoda [26] for counting colorings is an important technique that gets significantly below the number of colors required by Jerrum's approach. Extending Vigoda's approach to list *coloring* was first done by Chen et al. [10], roughly 20 years after Vigoda's breakthrough. Though they also surpassed a significant barrier at $11\Delta/6$ colors related to 1-step contractions in Hamming distance of two colorings that differ at a

vertex. The history of this problem suggests that further generalizations, e.g. to list packing may not be straightforward. While one can consider analogous dynamics for list packings and hope to reduce the bound on q in Theorem 1, we do not have results showing the existence of list packings in graphs of maximum degree Δ for fewer than $2\Delta - 2$ colors. In another direction, the use of more advanced Markov chain techniques to give perfect sampling of list packings could be interesting.

We finish with a natural conjecture on approximately counting list packings.

▶ **Conjecture 11.** *For each Δ and $q \geq 2\Delta$ there is an FPRAS for counting the number of q-list packings of graphs of maximum degree Δ.*

At the time of writing, we know of no reason that the lower bound on q cannot be reduced to, say, $\Delta + 1$. The value 2Δ represents a significant barrier in the sense that the existence of a list packing when $q \geq 2\Delta$ is an elementary consequence of Hall's theorem (though arguably not entirely trivial).

References

1 Noga Alon, Stijn Cambie, and Ross J. Kang. Asymmetric List Sizes in Bipartite Graphs. *Annals of Combinatorics*, 25(4):913–933, 2021. doi:10.1007/s00026-021-00552-5.

2 Ferenc Bencs, Ewan Davies, Viresh Patel, and Guus Regts. On zero-free regions for the anti-ferromagnetic Potts model on bounded-degree graphs. *Annales de l'Institut Henri Poincaré D*, 8(3):459–489, 2021. doi:10.4171/AIHPD/108.

3 Siddharth Bhandari and Sayantan Chakraborty. Improved bounds for perfect sampling of k-colorings in graphs. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pages 631–642, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3357713.3384244.

4 Peter Bradshaw. *Graph Colorings with Local Restrictions*. PhD thesis, Simon Fraser University, 2022. URL: https://summit.sfu.ca/item/35851.

5 R. Bubley and M. Dyer. Path coupling: A technique for proving rapid mixing in Markov chains. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 223–231, Miami Beach, FL, USA, 1997. IEEE Comput. Soc. doi:10.1109/SFCS.1997.646111.

6 Stijn Cambie, Wouter Cames van Batenburg, Ewan Davies, and Ross J. Kang. Packing list-colorings. *Random Structures & Algorithms*, 64(1):62–93, 2024. doi:10.1002/rsa.21181.

7 Stijn Cambie, Wouter Cames van Batenburg, and Xuding Zhu. Disjoint list-colorings for planar graphs. *arXiv preprint*, 2023. arXiv:2312.17233.

8 Stijn Cambie, Wouter Cames van Batenburg, Ewan Davies, and Ross J. Kang. List packing number of bounded degree graphs. *Combinatorics, Probability and Computing*, 33(6):807–828, 2024. doi:10.1017/S0963548324000191.

9 Charlie Carlson and Eric Vigoda. Flip Dynamics for Sampling Colorings: Improving $(11/6 - \epsilon)$ Using a Simple Metric. *To appear in SODA 2025*, 2025. doi:10.48550/arXiv.2407.04870.

10 Sitan Chen, Michelle Delcourt, Ankur Moitra, Guillem Perarnau, and Luke Postle. Improved Bounds for Randomly Sampling Colorings via Linear Programming. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, Proceedings, pages 2216–2234. Society for Industrial and Applied Mathematics, 2019. doi:10.1137/1.9781611975482.134.

11 Martin Dyer and Catherine Greenhill. A more rapidly mixing Markov chain for graph colorings. *Random Structures & Algorithms*, 13(3-4):285–317, 1998. doi:10.1002/(SICI)1098-2418(199810/12)13:3/4<285::AID-RSA6>3.0.CO;2-R.

12 Martin Dyer and Catherine Greenhill. Random Walks on Combinatorial Objects. In J. D. Lamb and D. A. Preece, editors, *Surveys in Combinatorics, 1999*, pages 101–136. Cambridge University Press, 1 edition, 1999. doi:10.1017/CBO9780511721335.005.

13 Paul Erdős, Arthur L. Rubin, and Herbert Taylor. Choosability in graphs. In *Proceedings of the West Coast Conference on Combinatorics, Graph Theory and Computing (Humboldt State Univ., Arcata, Calif., 1979)*, Congress. Numer., XXVI, pages 125–157. Utilitas Math., Winnipeg, Man., 1980.

14 P. Hall. On Representatives of Subsets. *Journal of the London Mathematical Society*, s1-10(1):26–30, 1935. doi:10.1112/jlms/s1-10.37.26.

15 Mark Huber. Exact sampling and approximate counting techniques. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 31–40, New York, NY, USA, 1998. Association for Computing Machinery. doi:10.1145/276698.276709.

16 Vishesh Jain, Ashwin Sah, and Mehtaab Sawhney. Perfectly Sampling $k \geq (8/3 + o(1))\Delta$-colorings in Graphs. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 1589–1600, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3406325.3451012.

17 Mark Jerrum. A very simple algorithm for estimating the number of k-colorings of a low-degree graph. *Random Structures & Algorithms*, 7(2):157–165, 1995. doi:10.1002/rsa.3240070205.

18 Mark Jerrum. *Counting, Sampling and Integrating: Algorithms and Complexity*. Lectures in Mathematics ETH Zürich. Birkhauser Verlag, Basel Boston, MA, 2003.

19 Mark R. Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986. doi:10.1016/0304-3975(86)90174-X.

20 Hemanshu Kaul, Rogers Mathew, Jeffrey A. Mudrock, and Michael J. Pelsmajer. Flexible list colorings: Maximizing the number of requests satisfied. *Journal of Graph Theory*, 106(4):887–906, 2024. doi:10.1002/jgt.23103.

21 Hemanshu Kaul and Jeffrey A. Mudrock. Counting Packings of List-colorings of Graphs. *arXiv preprint*, 2024. arXiv:2401.11025.

22 Jingcheng Liu, Alistair Sinclair, and Piyush Srivastava. A Deterministic Algorithm for Counting Colorings with 2-Delta Colors. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1380–1404, Baltimore, MD, USA, 2019. IEEE. doi:10.1109/FOCS.2019.00085.

23 Pinyan Lu and Yitong Yin. Improved FPTAS for Multi-spin Systems. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Prasad Raghavendra, Sofya Raskhodnikova, Klaus Jansen, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 8096, pages 639–654. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. doi:10.1007/978-3-642-40328-6_44.

24 Kyle MacKeigan. Independent coverings and orthogonal colourings. *Discrete Mathematics*, 344(8):112431, 2021. doi:10.1016/j.disc.2021.112431.

25 Jesús Salas and Alan D. Sokal. Absence of phase transition for antiferromagnetic Potts models via the Dobrushin uniqueness theorem. *Journal of Statistical Physics*, 86(3):551–579, 1997. doi:10.1007/BF02199113.

26 E. Vigoda. Improved bounds for sampling colorings. In *40th Annual Symposium on Foundations of Computer Science*, pages 51–59, New York City, NY, USA, 1999. IEEE Comput. Soc. doi:10.1109/SFFCS.1999.814577.

27 Vadim G Vizing. Coloring the vertices of a graph in prescribed colors. *Diskret. Analiz*, 29(3):10, 1976.

Locally Private Histograms in All Privacy Regimes

Clément L. Canonne
School of Computer Science, University of Sydney, Australia

Abigail Gentle
School of Computer Science, University of Sydney, Australia

Abstract

Frequency estimation, a.k.a. histograms, is a workhorse of data analysis, and as such has been thoroughly studied under differentially privacy. In particular, computing histograms in the *local* model of privacy has been the focus of a fruitful recent line of work, and various algorithms have been proposed, achieving the order-optimal ℓ_∞ error in the high-privacy (small ε) regime while balancing other considerations such as time- and communication-efficiency. However, to the best of our knowledge, the picture is much less clear when it comes to the medium- or low-privacy regime (large ε), despite its increased relevance in practice. In this paper, we investigate locally private histograms, and the very related distribution learning task, in this medium-to-low privacy regime, and establish near-tight (and somewhat unexpected) bounds on the ℓ_∞ error achievable. As a direct corollary of our results, we obtain a protocol for histograms in the *shuffle* model of differential privacy, with accuracy matching previous algorithms but significantly better message and communication complexity.

Our theoretical findings emerge from a novel analysis, which appears to improve bounds across the board for the locally private histogram problem. We back our theoretical findings by an empirical comparison of existing algorithms in all privacy regimes, to assess their typical performance and behaviour beyond the worst-case setting.

2012 ACM Subject Classification Security and privacy; Security and privacy → Usability in security and privacy; Security and privacy → Privacy protections; Theory of computation → Theory of database privacy and security

Keywords and phrases Differential Privacy, Local Differential Privacy, Histograms, Frequency Estimation, Lower Bounds, Maximum Error

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.25

Related Version *Full Version*: https://arxiv.org/abs/2408.04888

Funding *Clément L. Canonne*: Supported by an ARC DECRA (DE230101329).

Acknowledgements The authors would like to thank Guy Blanc for the proof of Lemma 17, and Albert Cheu for insightful discussions regarding the use of amplification by shuffling (Theorem 25). This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing.

1 Introduction

Frequency estimation is a fundamental problem of statistics: besides its use for basic surveying, it is also used as a building block in distribution learning, identifying heavy hitters in sparse domains, and regression, correlation or covariance analysis. As such, frequency estimation (and the closely related problem of heavy hitters) have been thoroughly studied in a variety of settings, ranging from streaming to privacy-preserving analytics. We here focus on the latter, and specifically on the *local* model of differential privacy (LDP), where the data is distributed across a large number of users, and each datum is subject to stringent differential privacy requirements. This question has, over the past decade, received a lot of attention, starting with [8, 28]; as we elaborate in Section 1.1, much is known about locally private frequency


© Clément L. Canonne and Abigail Gentle;
licensed under Creative Commons License CC-BY 4.0



estimation, to the point that it may seem the question has been, already fully resolved – both in theory and practice. However, most recent large-scale implementations of local privacy (see, e.g., [5, 18]) must balance many efficiency objectives, including the bandwidth and computational requirements (as, among others, a proxy for energy consumption), and, above all, the estimation accuracy. Keeping this accuracy in check has, in turn, led to the common use of relatively large values of the "privacy parameter" ε, far from the original rule of thumb of $\varepsilon \ll 1$.[1]

In view of the relevance of frequency estimation to privacy-preserving algorithms and deployments, it is crucial that theory be informed by, and applicable to practice; and that constant factors, bounds on achievability, and the behaviour of estimation error in all parameter regimes are well understood. Yet, to the best of our knowledge, most if not all of the previous theoretical work on locally private frequency estimation focuses on the high-privacy $\varepsilon \ll 1$ regime, leaving the low-privacy regime by and large uncharted. Addressing this gap in our theoretical understanding is the focus of our work.

The question and setting

We will focus on examining the standard worst-case estimation error, or the ℓ_∞-error as it is most relevant to the problem, with special importance for controlling error rates in heavy hitters. Specifically, when estimating the empirical frequencies $q = (q_1, \ldots, q_k)$ over a universe $\mathcal{X}$ of size k from n users, the (expected) error is given by

$$\mathbb{E}[\|\hat{q} - q\|_\infty] = \mathbb{E}\left[\max_{1 \le i \le k} |\hat{q}_i - q_i|\right] \tag{1}$$

where the expectation is taken over the (possibly randomised) algorithm given as input the users' data, and $\hat{q}$ is the vector of estimated frequencies. We are interested in the worst-case error over all possible datasets, that is, the supremum of the above quantity over all inputs $X := (X_1, \ldots, X_n) \in \mathcal{X}^n$ (equivalently, all q). We will seek algorithms ("protocols") to minimize this worst-case error under the constraint of (pure) local privacy (see Section 2), parameterized by the privacy parameter $\varepsilon > 0$. As in most distributed settings, one can consider several variants depending on whether a common random seed is available to the users (public-coin protocols) or not (private-coin protocols). It is known that for locally private frequency estimation allowing public-coin protocols can provably reduce the communication requirements [3];[2] similarly, one can allow some back and forth between users and the central server (interactive protocols). However, the public-coin or interactive settings come at an increased deployment costs, and are often less easy (or even impossible) to implement as they require either broadcasting from the center or sustained two-way communication between the parties. As our focus is chiefly on analyzing the most versatile setting, we hereafter restrict ourselves for our algorithms to the private-coin (non-interactive) setting; however, we mention that our lower bounds apply to the public-coin setting as well.

A closely related question is that of *distribution learning* under ℓ_∞ loss, which differs from frequency estimation in that the dataset X is not arbitrary, but instead assumed to be drawn i.i.d. from an unknown probability distribution q over $\mathcal{X}$: in this sense, distribution learning

[1] See, for instance, the list of real-case use of differential privacy maintained here, which includes use cases of LDP along with the correspond stated values of ε: `https://desfontain.es/blog/real-world-differential-privacy.html`.

[2] Specifically, the cited paper establishes a separation between public- and private- coin protocols, showing that the former can achieve vanishing error even with constant-length communication per user, a setting where the latter must incur $\Omega(1)$ error. To complement this, it is known that using public randomness one can always reduce the per-player communication to $\lceil \varepsilon \rceil$ bits [8].

is an "easier" problem than frequency estimation, and an algorithm for the latter implies one for the former.[3] Formal definitions, as well as the relation between these two questions, can be found in Section 2. In this paper, we therefore focus on frequency estimation, and will point out the implications for distribution learning as corollaries.

Connection to shuffle privacy

To conclude this subsection, we mention that besides the use of large values of ε in practical deployments of locally private algorithms, which calls for a better understanding of the tasks in this parameter regime, another motivation for studying the "low-privacy regime" comes from the emergence of another model of differential privacy, *shuffle privacy* [15, 21], and its increasingly widespread adoption. Indeed, it is known that one generic way to obtain shuffle private algorithms is via the so-called "amplification-by-shuffling" technique (see, e.g., [23]), whereby a locally private algorithm with high ε (low privacy) yields a shuffle private protocol with small ε (high privacy), and the same number of messages and communication cost. This makes characterizing the low-privacy regime of locally private algorithms a very consequential question, especially for fundamental tasks such as frequency estimation and the related distribution learning.

1.1 Prior work

Many innovative algorithms have been proposed in recent years for locally private histograms and distribution estimation [22, 3, 14, 34, 33, 25, 24], with a subset focusing on ℓ_∞-error.[4] A lower bound of $\ell_\infty = \Omega(\sqrt{\log k/\varepsilon^2 n})$ for frequency estimation under local differential privacy was established by [8] for most regimes of k, n, and small privacy parameter ε; and various LDP protocols asymptotically achieving this bound have been proposed (see Table 1).

Most recently, upper bounds on ℓ_∞ for frequency estimation were derived in [14], and evaluated empirically in [24]. For large ε, i.e., the "low-privacy regime," to the best of our knowledge the best current results are (1) a bound of $O(\sqrt{\log k/\varepsilon n})$ on the ℓ_∞ error rate established in [14], and, (2) an upper bound of $O(\sqrt{\log k/(e^{\varepsilon/2} n)} + (\log k)/n)$ from [29] (Theorem 2), based on CountSketch: note that this error does not vanish as ε grows.

In [25], it was stated that under most commonly used metrics $(\ell_\infty, \ell_2^2, \ell_1)$, the error is primary driven by the variance of an algorithm. We show that at least for ℓ_∞ a close look at sub-Gaussian, and sub-gamma behaviour can better capture the error in low-privacy regimes. Additionally, [14] raise the question of whether the upper bound of $O(\sqrt{\log k/\varepsilon n})$, is tight in the low privacy regime. We show in Theorem 14 that, quite surprisingly, it is not. Histograms have also seen detailed analysis in the shuffle model as both a practical question [26, 27, 15, 16, 9] and as a benchmark for reasoning about the power of the shuffle model [6]. As two points of comparison we highlight [26] which introduced a multiple–message protocol for shuffled histograms achieving expected ℓ_∞-error $O(\sqrt{\log k \log(1/\delta)}/(\varepsilon n))$ with $O(k^{1/100})$ rounds of communication, and [6] which demonstrated expected maximum error of $O(\log(1/\delta)/(\varepsilon^2 n))$ with $k+1$ rounds. While our results do not achieve the latter bound, we show surprisingly that the former can be achieved in some fairly permissive parameter regimes with only a single message.

[3] There are some subtleties here, but it is worth noting, as a baseline, that absent privacy constraints it is well-known that the error of learning to ℓ_∞ scales as $1/\sqrt{n}$ with no dependence on the alphabet size k; while frequency estimation can be done with zero error, as each user can simply send their data.

[4] For distribution estimation, a common error measure in the literature is ℓ_1, corresponding to total variation distance.

Table 1 Selection of best known upper bounds for communication, ℓ_∞ and ℓ_2^2-error, where the shaded cells are our results († *indicates results not established in the corresponding paper, but that we derive for completeness.*) As discussed above, "private-coin" refers to the fact that the users do not require access to a common random seed (typically easier to implement than "public-coin" protocols, where they do). We note that all public-coin protocols mentioned here can be made private-coin at the cost of an $O(\log k)$ blowup factor in the communication cost.

Protocol	Private-coin	Communication	ℓ_∞ error	ℓ_2^2 error
RAPPOR	✓	k	$\sqrt{\frac{\log k}{n\min(1,\varepsilon^2)}}$ $\sqrt{\frac{\log k}{n\min(\varepsilon,\varepsilon^2)}}$ $\sqrt{\frac{\log k}{ne^{\varepsilon/2}}} + \frac{\log k}{n\varepsilon}\cdot\log n$	$\frac{ke^{\varepsilon/2}}{n\left(e^{\varepsilon/2}-1\right)^2}$
Subset Selection [33]	✓	$\frac{k}{e^\varepsilon}\max(1,\varepsilon)$	?	$\frac{ke^\varepsilon}{n(e^\varepsilon-1)^2}$
(G)HR [3, Theorem 7]	✓	$\log k$	?	$\frac{ke^\varepsilon}{n(e^\varepsilon-1)^2}$
RHR [14, Theorem 3.1]	×	ℓ	?	$\frac{k}{n\min\left(\left(e^{\varepsilon/2}-1\right)^2,e^\varepsilon,2^\ell,k\right)}$
No name [14, Theorem 3.1]	×	ℓ	$\sqrt{\frac{\log k}{n\min(\varepsilon,\varepsilon^2,\ell)}}$	$\frac{ke^{2\varepsilon/\ell}}{n\ell\left(e^{\varepsilon/\ell}-1\right)^2}$ (†)
CountSketch-based [29]	×	$\max(\varepsilon,\log\frac{1}{\varepsilon})$	$\sqrt{\frac{\log k}{n(e^{\varepsilon/2}-1)^2}} + \frac{\log k}{n}$	$\frac{ke^{\varepsilon/2}}{n\left(e^{\varepsilon/2}-1\right)^2}$
PGR [24]	✓	$\log k$	$\sqrt{\frac{\log k}{ne^\varepsilon}} + \frac{\log k}{n\varepsilon}\cdot\log n$	$\frac{ke^\varepsilon}{n(e^\varepsilon-1)^2}$

1.2 Overview of results

We here summarise our main results, and briefly discuss the underlying techniques. While we focus on two protocols in particular, the techniques themselves are broadly applicable. Our first set of results concerns the ℓ_∞ error rate achievable by LDP protocols for frequency estimation. We first recall, as a baseline, a general transformation which converts any LDP protocol with optimal error in the high-privacy regime into another LDP protocol with *reasonable error* in the low-privacy regime, at the cost of a blowup in communication:

▶ **Proposition 1** (Informal; see Proposition 9)**.** *Let A be any locally private protocol for frequency estimation with expected ℓ_∞ error $O\left(\sqrt{\log k/(n\varepsilon^2)}\right)$ for $\varepsilon \le 1$, using ℓ bits of communication per user. Then there is a locally private protocol A' achieving error*

$$O\left(\sqrt{\frac{\log k}{n\min(\varepsilon,\varepsilon^2)}}\right)$$

for all $\varepsilon > 0$, using $\ell\lceil\varepsilon\rceil$ bits of of communication per user.

We emphasise that this transformation is not new, and mimics an argument found in, e.g., [14]. This general transformation is quite appealing, as it provides (in theory) good performance in the low-privacy regime "for free" given any good enough LDP protocol for the high-privacy one. However, this comes at a price: first, the communication blowup, which could be impractical; second, a loss in constant factors, which while relatively small might still be prohibitive; and, perhaps more importantly, *this requires changing the existing algorithm* (and as a result the data analysis pipeline), which is often a significant hurdle. Our

second result, focusing on one of the earliest, versatile, and (at least in its "vanilla" version) conceptually simple LDP protocols for frequency estimation, RAPPOR [22], shows that this transformation is not actually necessary, and that RAPPOR actually achieves this improved bound *without* modification:

▶ **Theorem 2** (Informal; see Theorem 13). *The (simple version of) RAPPOR achieves expected ℓ_∞ error*

$$O\left(\sqrt{\frac{\log k}{n \min(\varepsilon, \varepsilon^2)}}\right)$$

for all $\varepsilon > 0$, using k bits of of communication per user.

In comparison, using the generic transformation above on RAPPOR would require $\lceil \varepsilon \rceil$ k-bit messages per user, and, in terms of worst-case theoretical bounds, an expected error worse by a factor $\simeq 2.04$. As such, our first result can be summarized as saying that *analyzing (again) an existing algorithm can be better than modifying it* – and, quite importantly, that it may not be necessary to change an existing algorithmic pipeline to inherit better guarantees.

The proof of the above result relies on a careful analysis of the expected maximum of sums of Bernoulli random variables, and specifically on a fine-grained analysis of their subgaussian behaviour in the "highly biased" regime. While RAPPOR is particularly amenable to this analysis, we believe that this technique is highly general and applicable to a broad range of LDP protocols, for example those following the general "scheme template" of [4].

Yet, trying to establish optimality of this $\min(\varepsilon, \varepsilon^2)$ scaling turned out to be very challenging. And indeed, *this is for a good reason:* as we show, it is actually possible to achieve significantly better error rate in the low-privacy regime – and, surprisingly, this much better error is again attained by RAPPOR, out-of-the-box:

▶ **Theorem 3** (Informal; see Theorem 14). *The (simple version of) RAPPOR achieves expected ℓ_∞ error*

$$\tilde{O}\left(\max\left(\sqrt{\frac{\log k}{n e^{\varepsilon/2}}}, \frac{\log k}{n\varepsilon}\right)\right)$$

for all $\varepsilon \geq 1$, using k bits of of communication per user.

The $\tilde{O}$ notation here only hides a single $\log n$ factor in the second term. As the ℓ_∞ error is always at most 1, up to this $\log n$ factor this is always better than Theorem 2.

The proof of this result requires going beyond the sub-gaussian concentration behaviour alluded to before, and instead analyse the maximum of these sums of Bernoulli random variables as *sub-gamma* random variables. More precisely, we draw upon very recent work by [17] (see also [10]) on "local Glivenko–Cantelli" results, which provide refined concentration bounds for mean estimation of high-dimensional product distributions – in our case, the distributions over $\{0,1\}^k$ arising from the use of RAPPOR.

The above result is appealing, in that it not only yields better error rate than previously known (or, in many cases, believed to be possible) in the $\varepsilon \gg 1$ regime; but also in that it is achieved "for free" by an existing and widely used algorithm. However, it does have one negative aspect: namely, that RAPPOR is, in its standard version, very inefficient from a communication point of view, as it require one k-bit message from each user – far from the ideal $O(\log k)$ bits one could hope for. Luckily, as mentioned above, the analysis underlying Theorem 3 is quite general, and applies to a broad range of locally private

estimation algorithms. While it does not lead to the improved bound for *all* such protocols, we show that it applies, for instance, to the recently proposed Projective Geometry Response protocol of Feldman, Nelson, Nguyen, and Talwar [24], whose performance was originally only analyzed for ℓ_2 error:

▶ **Theorem 4** (Informal; see Theorem 18)**.** *Projective Geometry Response (PGR) achieves expected ℓ_∞ error*

$$\tilde{O}\left(\max\left(\sqrt{\frac{\log k}{ne^\varepsilon}}, \frac{\log k}{n\varepsilon}\right)\right)$$

for all $\varepsilon \geq 1$, using $O(\log k)$ bits of of communication per user.

Note that again this is achieved "out-of-the-box" by an existing algorithm, without any modification! As a result, this inherits all of the features of PGR its authors originally established: crucially, its computational efficiency. We also point out that the stated guarantee is even slightly better than that of RAPPOR, as the first term now features an $e^{-\varepsilon}$ dependence (instead of the $e^{-\varepsilon/2}$ of Theorem 3).

We then turn to prove optimally of this error rate, and show that this is, up to a logarithmic factor, optimal for any LDP protocol:

▶ **Theorem 5** (Informal; see Theorem 24)**.** *Any LDP protocol for frequency estimation must have, in the worst case, expected ℓ_∞ error*

$$\Omega\left(\sqrt{\frac{\log k}{n\varepsilon^2}}\right)$$

for $\varepsilon \in (0, 1]$, and

$$\Omega\left(\max\left(\sqrt{\frac{\log k}{ne^\varepsilon}}, \frac{\log k}{n\varepsilon}\right)\right)$$

for $\varepsilon \geq 1$.

We remark that this lower bound also applies to the "easier" question of distribution learning, as we will state momentarily. The first part of this lower bound, as mentioned before, was already known; the second part is new, and, while not necessarily difficult to show in hindsight, does require significant care in combining inequalities between various information-theoretic quantities to avoid ending up with a vacuous bound. In this sense, our main contribution for the lower bound is to establish it in a self-contained, streamlined fashion, by drawing on the "chi-square contraction" framework of [2]; and – importantly – that it matches the upper bound obtained earlier in *both* the high- and low-privacy regimes. Another interesting aspect of this lower bound is that all three terms are derived separately, but from the same family of hard instances: a dataset where almost all users hold the same element.

Implications for shuffle privacy

By combining our new results on LDP protocols for histograms in the low-privacy regime with known "amplification-by-shuffling" results, we are able to obtain a simple, robust shuffle (approximate) DP protocol using only *one* message per user of logarithmic length, while achieving state-of-the-art:

▶ **Theorem 6** (Informal; see Theorem 27). *Shuffling the Projective Geometry Response protocol achieves expected ℓ_∞ error*

$$O\left(\frac{\sqrt{\log(k)\log(1/\delta)}}{n\varepsilon}\right)$$

for all $\varepsilon \in [\Omega(1/\sqrt{n}), 1]$, with $O(\log k)$ bits of communication and a single message per user.

This matches that of the best known shuffle DP protocols, but with a much lower message and communication complexity. It was, to the best of our knowledge, still open whether achieving these "best of three worlds" guarantees was possible.

We conclude by providing the corollary of our results for distribution learning, using the known connection to frequency estimation (along with the standard lower bound of $\Omega(1/\sqrt{n})$ on the error rate for the question, absent privacy constraints):

▶ **Corollary 7.** *For distribution learning under local privacy constraints, the minmax ℓ_∞ error achievable is at most*

$$\tilde{O}\left(\max\left(\sqrt{\frac{\log k}{ne^\varepsilon}}, \frac{\log k}{n\varepsilon}, \frac{1}{\sqrt{n}}\right)\right)$$

for all $\varepsilon \geq 1$, and

$$O\left(\sqrt{\frac{\log k}{n\varepsilon^2}}\right)$$

for $\varepsilon \in (0, 1]$; and this is tight, up to a logarithmic factor (in n). Moreover, the upper bound is attained by PGR, using $O(\log k)$ bits of of communication per user.

Organisation

After providing some background and setting notation, we establish our theoretical results (upper bounds on the error rate) in Section 3, followed by information-theoretic lower bounds in Section 4 and applications to shuffle privacy in Section 5. Finally, we discuss our results and potential future work in Section 6. Several proofs and a discussion of empirical analyses – omitted for clarity of exposition – can be found in the full version.

2 Preliminaries and notation

(Local) differential privacy

We first recall the definition of local differential privacy (LDP):

▶ **Definition 8** (Locally private randomiser). *An algorithm $Q\colon \mathcal{X} \to \mathcal{Y}$ satisfies ε-differential privacy if for all pairs of inputs $x, x' \in \mathcal{X}$, all sets of outputs $S \subseteq \mathcal{Y}$ are ε-close,*

$$\Pr[\,Q(x) \in S\,] \leq e^\varepsilon \Pr[\,Q(x') \in S\,].$$

A *locally private protocol* is typically a pair of algorithms $Q\colon \mathcal{X} \to \mathcal{Y}$, executed by the user on their data, and $A\colon \mathcal{Y}^n \to \mathcal{X}^k$, executed by the server to transform the randomised outputs into an unbiased estimator for the true quantity. It is well known (see, e.g., [19]) that all LDP Frequency Estimation algorithms have a stochastic matrix representation which maps every element in the input alphabet to a probability distribution over the output

alphabet. Furthermore, [30] demonstrated that all optimal mechanisms obey a "binary" property, in that their stochastic matrix only contains probabilities weighted by either e^{ε} or 1 (appropriately normalised). Recently another "family of LDP protocols" was introduced by [4] and used in [24], where each input is associated with a set of high probability outputs determined by some useful set system.

Frequency Estimation

Under frequency estimation, the aim is to estimate the empirical frequency of the observations. We denote the dataset of n observations as $X^n \coloneqq \{x_1, \ldots, x_n\}$ and their empirical frequencies as the probability vector $q \coloneqq q(X^n)$, where $q_i \coloneqq \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}_{\{x_j=i\}}$. That is, q is an element of the probability simplex,

$$\Delta_k \coloneqq \left\{ p \in \mathbb{R}^k_{\geq 0} : \sum_{i=1}^{k} p_i = 1 \right\}$$

A *frequency estimator* over an alphabet $[k]$ is a (possibly randomized) function $\mathcal{A}\colon [k]^n \to \Delta_k$, which approximates the true frequencies sufficiently well. The quality of the resulting approximation, denoted $\hat{q}$, is measured through a suitably chosen loss function. Typical choices of distance are the ℓ_r norms, for $r \in [1, \infty]$; in this paper, we will be mostly concerned with the ℓ_∞ distance, where $d(q, \hat{q}) = \|q - \hat{q}\|_\infty = \max_{1\leq i\leq k} |q_i - \hat{q}_i|$, and consider the *expected loss* $\mathbb{E}[\|q - \hat{q}\|_\infty]$ (where the expectation is over the randomness of the frequency estimator).

Distribution learning (estimation)

Given independent and identically distributed (i.i.d.) samples from an unknown probability distribution p, the goal of distribution learning (density estimation) is to find a distribution $\hat{p}$ that approximates p sufficiently well. As before, the quality of the approximation is typically measured through a suitably chosen loss function, quantifying the distance between the true distribution p and its approximation $\hat{p}$. From the above, one can see that the key difference between frequency and distribution (under the same loss function) is that in the former, observations are arbitrary, while in the latter they are assumed i.i.d. from some common underlying probability distribution. As such, the expected loss also takes into account the randomness of the samples themselves.

Frequency Estimation implies Distribution Estimation

For any given distance measure d we denote the expected sampling error (under loss d) between the empirical histogram $\hat{p}_n$ from n i.i.d. samples and the true underlying distribution p as $\Phi(d, k, n)$:

$$\Phi(d, k, n) = \sup_{p\in\Delta_k} \mathbb{E}[d(p, \hat{p}_n)]$$

Further, denote the optimal worst-case (over p) expected error under loss d from n i.i.d. samples under ε-LDP constraints as $\Phi^{\text{priv}} \coloneqq \Phi^{\text{priv}}(d, \varepsilon, k, n)$. The "optimal" here refers to quantifying over all locally private distribution estimators. We can analogously express the optimal expected loss when learning the empirical frequencies as $\varphi^{\text{priv}} \coloneqq \varphi^{\text{priv}}(d, \varepsilon, k, n)$.

▶ **Fact 1.** *Accurate* frequency *estimation implies accurate* distribution *estimation: namely,* $\Phi^{\text{priv}}(d, \varepsilon, k, n) \leq \Phi(d, k, n) + \varphi^{\text{priv}}(d, \varepsilon, k, n)$.

For specific norms this allows us to derive (1) a lower bound on φ^{priv} (locally private frequency estimation) from a lower bound on Φ^{priv} (locally private distribution estimation), and (2) an upper bound on Φ^{priv} from an upper bound on φ^{priv}, recalling that (see, for instance, [31, 13])

$$\Phi(d,k,n) = \begin{cases} \Theta\left(\sqrt{\frac{k}{n}}\right) & \text{when } d = \ell_1 \\ \Theta\left(\frac{1}{\sqrt{n}}\right) & \text{when } d = \ell_2, \ell_\infty \end{cases}$$

3 Better algorithms in the low-privacy regime

Although the order-optimal ℓ_∞ error rates for LDP frequency estimation and distribution learning are well-understood by now in the high-privacy regime, with many distinct algorithms achieving the tight

$$O\left(\sqrt{\frac{\log k}{n\varepsilon^2}}\right)$$

expected error bound, much less is understood about the best achievable error for large, or even medium, values of ε. In this section, we first revisit (and slightly generalize) an idea from [14], which shows how to convert any protocol "optimal in the high-privacy regime" to a related protocol "good enough in the low-privacy regime as well", at the price of a blowup in communication (Section 3.1). We then focus on the specific example of RAPPOR, showing that – perhaps surprisingly – this simple protocol does already achieve the same bound by *without any modification* nor communication blowup (Section 3.2). We finally show that for distribution learning, this very same RAPPOR in fact achieves an *even better* error rate than this, leveraging very recent results on concentration of the empirical mean for high-dimensional distributions, due to [17, 10] (Section 3.3).

3.1 A generic transformation, and a baseline

Here we prove the following generic statement:

▶ **Proposition 9.** *Suppose there exists a symmetric*[5] *LDP protocol A for frequency estimation achieving expected ℓ_∞ error*

$$\mathcal{E}(n,k,\varepsilon) = O\left(\sqrt{\frac{\log k}{n\min(1,\varepsilon^2)}}\right)$$

for $\varepsilon \le 1$, with m messages and ℓ bits of communication per user. Then, for every integer $L \ge 1$, there exists an LDP protocol A' for frequency estimation achieving expected ℓ_∞ error

$$\mathcal{E}\left(n\cdot\min(\lceil\varepsilon\rceil, L), k, \frac{\varepsilon}{\min(\lceil\varepsilon\rceil, L)}\right) = O\left(\sqrt{\frac{\log k}{n\min(L,\varepsilon,\varepsilon^2)}}\right)$$

for all $\varepsilon > 0$, with $m\min(\lceil\varepsilon\rceil, L)$ messages and $\ell\min(\lceil\varepsilon\rceil, L)$ bits of communication per user. Further, if A is a public-coin (resp. private-coin) protocol, then so is A'.

[5] i.e., where all users use the same randomiser.

The idea behind this result is not new, and is used (in a slightly less general form) in [14, Appendix E.4]. We restate and prove it in this paper in a self-contained form, as we believe it to be of broader interest.

As a direct corollary (setting $L = \lceil \varepsilon \rceil$), applying the above to Hadamard Response and RAPPOR, for instance, we obtain the following bounds:

▶ **Corollary 10.** *For every $\varepsilon > 0$, there exists a private-coin ε-LDP protocol (namely, a modification of Hadamard Response) for frequency estimation with expected ℓ_∞ error*

$$O\left(\sqrt{\frac{\log k}{n \min(\varepsilon, \varepsilon^2)}}\right)$$

using $\lceil \varepsilon \rceil$ messages per user, and $\log k + O(1)$ bits per message.

▶ **Corollary 11.** *For every $\varepsilon > 0$, there exists a private-coin ε-LDP protocol (namely, a modification of RAPPOR) for frequency estimation with expected ℓ_∞ error*

$$O\left(\sqrt{\frac{\log k}{n \min(\varepsilon, \varepsilon^2)}}\right)$$

using $\lceil \varepsilon \rceil$ messages per user, and k bits per message.

3.2 Tighter analysis for RAPPOR

To do so, we first recall some facts and notation about RAPPOR, which will help with the analysis of its performance. A common simplification of the RAPPOR protocol [4, 3] is to parameterise it by ε as follows.

1. Given input $x_i \in [k]$, one-hot encode it onto a k-bit vector e_x s.t. only the x'th bit is 1.
2. Flip each bit independently with probability $\frac{1}{e^{\varepsilon/2}+1}$ and send the resulting noisy bit-vector Y_i to the server.[6]
3. The server then receives n noisy bit vectors, computes $\bar{Y} \leftarrow \frac{1}{n}\sum_{i=1}^{n} Y_i$, and estimates

$$\hat{q} = \frac{e^{\varepsilon/2}+1}{e^{\varepsilon/2}-1}\bar{Y} - \frac{1}{e^{\varepsilon/2}-1}\mathbf{1}_k \tag{2}$$

where $\mathbf{1}$ is the 1-vector of size k.

A standard fact, which we recall here for completeness, is that $\hat{q}$ defined above is an unbiased estimator for the true vector of frequencies q:

▶ **Lemma 12** (Expectation of $\hat{q}$)**.** *We have $\mathbb{E}[\hat{q}] = q$.*

We then have $\mathbb{E}[\|\hat{q} - q\|_\infty] = \frac{e^{\varepsilon/2}+1}{e^{\varepsilon/2}-1}\mathbb{E}\left[\|\bar{Y} - \mathbb{E}\left[\bar{Y}\right]\|_\infty\right]$, and so to bound the expected ℓ_∞ error is suffices to bound that of the expected maximum deviation of the $\bar{Y}_j$'s from their mean. Now, each $\bar{Y}_j$ is the (normalised) sum of n independent Bernoulli random variables: the standard way to analyse this maximum is to recall that a sum of n independent Bernoullis is a sub-gaussian random variable with parameter[7] at most $\frac{n}{4}$. Standard results (see, for example [32, Chapter 2]) on the maximum of k (not necessarily independent) sub-Gaussian random variables then give

$$\mathbb{E}[\|\hat{q} - q\|_\infty] \leq \frac{e^{\varepsilon/2}+1}{e^{\varepsilon/2}-1}\sqrt{\frac{\log k}{2n}}$$

[6] This is just the "Permanent Randomised Response" step of RAPPOR parameterised by ε.

[7] Importantly, note that this may not coincide with the variance of X, although it does in some important cases (e.g., for a Gaussian r.v.).

which indeed behaves as $O(\sqrt{\log k/(n\varepsilon^2)})$ for small ε. However, the bound quickly degrades for large ε, and only yields $O(\sqrt{\log k/n})$ (no dependence on the privacy parameter at all!) as ε grows.

To remedy this, we will need a tighter analysis of the subgaussian parameter of Bernoulli random variables in the "very biased" regime (which is the one we have to handle for large ε, as then $\frac{1}{e^{\varepsilon/2}+1} \approx 0$). Specifically, we will rely on the following characterisation of the sub-gaussian norm $\sigma^2(p)$ of a (centered) Bern(p) random variable, known as the Kearns–Saul inequality, and which can be found in, e.g., [12]:

$$\sigma^2(p) = \begin{cases} 0, & p \in \{0,1\} \\ \frac{1}{4}, & p = \frac{1}{2} \\ \frac{2p-1}{2\log\frac{p}{1-p}} \end{cases} \tag{3}$$

Importantly, this expression is symmetric: $\sigma^2(p) = \sigma^2(1-p)$ for all $p \in [0,1]$. In our setting, each Y_j is the sum of n independent Bernoulli random variables with one of two symmetric parameters, $p := \frac{1}{e^{\varepsilon/2}+1}$ or $1-p = \frac{e^{\varepsilon/2}}{e^{\varepsilon/2}+1}$. The sub-Gaussian norm at play is then $\sigma^2(p) = \frac{e^{\varepsilon/2}-1}{(e^{\varepsilon/2}+1)\varepsilon}$. Using sub-additivity of the sub-gaussian parameter for independent random variables, we can bound the sub-Gaussian norm of Y_j as

$$\sigma^2(Y_j) \leq \frac{(e^{\varepsilon/2}-1)}{(e^{\varepsilon/2}+1)n\varepsilon} \tag{4}$$

We can use this to bound the maximum error of the estimator.

▶ **Theorem 13** (Expected maximum error of RAPPOR). *The expected maximum of $\hat{q}$ is given by*

$$\mathbb{E}[\|\hat{q}-q\|_\infty] \leq \sqrt{\frac{2\left(e^{\varepsilon/2}+1\right)\log k}{n\left(e^{\varepsilon/2}-1\right)\varepsilon}} \in O\left(\sqrt{\frac{\log k}{n\min(\varepsilon,\varepsilon^2)}}\right) \tag{5}$$

proving the main result of this subsection.

3.3 Optimal analysis for RAPPOR

It is natural to wonder whether this $\min(\varepsilon,\varepsilon^2)$ dependence on the privacy parameter is order-optimal; especially as it appears in other locally private estimation tasks, such as mean estimation for high-dimensional Gaussians or product distributions [20, 1]. Quite surprisingly, we will show that for frequency estimation the ℓ_∞ error rate given in Equation (5) is *not* optimal for large values of ε, and that even the simple RAPPOR algorithm can achieve significantly better.

▶ **Theorem 14.** *For $\varepsilon \geq 1$, the expected ℓ_∞ error of RAPPOR for frequency estimation satisfies*

$$\mathbb{E}[\|\hat{q}-q\|_\infty] = O\left(\sqrt{\frac{\log k}{ne^{\varepsilon/2}}} + \frac{\log k}{n\varepsilon}\cdot\log n\right).$$

Importantly, this is better than the bound in Equation (5) whenever $n = \tilde{\Omega}(\log k/\varepsilon)$, which is the regime of interest (small constant, or vanishing, error). To see why this better bound may hold for large ε, recall that the bound given in Equation (5) relies on analyzing the

expected maximum of k (centered) Binomials random variables using their *sub-gaussian* behavior. This is good when the parameters of the Binomials are not too skewed; however, in the low privacy regimes, the parameters of the Bernoulli summands become very close to 0 (or 1): in that case, to analyze the expected maximum of the Binomials it is tighter to see them as having a *sub-gamma* behavior. Details follow.

Proof of Theorem 14. As alluded to above, we want to analyze the expected behavior of the maximum of Binomial random variables *beyond* the sub-gaussian regime. Invoking generic bounds for sub-gamma random variables such as [11, Corollary 2.6], unfortunately, does not lend itself to the order-optimal bounds either. Instead, we rely on the "local Glivenko–Cantelli" bounds recently obtained by [17, 10], which provide a more refined upper bound: to introduce the result we will invoke, we first need some notation.

▶ **Definition 15.** *Given $\mu \in [0,1]^{\mathbb{N}}$, denote by $\tilde{\mu} \in [0,1/2]^{\mathbb{N}}$ the sequence defined by $\tilde{\mu}_i = \min(\mu, 1-\mu)$ for all $i \geq 1$, and by $\tilde{\mu}^{\downarrow}$ its non-increasing rearrangement. Finally, let p_μ denote the product distribution over $\{0,1\}^{\mathbb{N}}$ with mean vector μ.*

With this in hand, the main result of [17] can be restated as follows:

▶ **Theorem 16** ([17, Theorem 3])**.** *Let $n \geq 21$, and suppose that $\mu \in [0,1]^{\mathbb{N}}$ is such that $\lim_{i\to\infty} \tilde{\mu}_i^{\downarrow} = 0$. Let $\hat{\mu}_n = \frac{1}{n}\sum_{j=1}^{n} X_j \in [0,1]^{\mathbb{N}}$ denote the empirical estimator for μ given n i.i.d. samples $X_1, \dots, X_n$ from p_μ. Then the following holds:*

$$\mathbb{E}[\|\hat{\mu}_n - \mu\|_\infty] \lesssim \sup_{i\geq 1} \sqrt{\frac{\tilde{\mu}_i^{\downarrow} \log(i+1)}{n}} + \frac{\log n}{n} \sup_{i\geq 1} \frac{\log(i+1)}{\log \frac{1}{\tilde{\mu}_i^{\downarrow}}}$$

(Note that [10] recently improved on this results by up to a $\log n$ factor in the second term. For simplicity, we use the slightly weaker result of [17], as it is easier to manipulate.)

We want to apply this result to bounding $\mathbb{E}\big[\|\bar{Y} - \mathbb{E}\big[\bar{Y}\big]\|_\infty\big]$. However, we face one obstacle in doing so, as we do not have a product distribution over $\{0,1\}^k$: while $\bar{Y}_1, \dots, \bar{Y}_k$ are indeed independent, and each of the form

$$\bar{Y}_i = \frac{1}{n}\sum_{j=1}^{n} Y_{i,j}$$

where $Y_{i,1}, \dots, Y_{i,n}$ are independent Bernoulli random variables, these Bernoullis are not identically distributed: exactly $n_i = nq_i$ of them have parameter $1-p = \frac{e^{\varepsilon/2}}{e^{\varepsilon/2}+1}$, and the remaining $n - n_i$ have parameter $p = \frac{1}{e^{\varepsilon/2}+1}$. Of course, we do not know the n_i's, as this is what we are trying to estimate; all we have is that

$$\sum_{i=1}^{k} n_i = n\,.$$

To circumvent this issue, let us write, for each $1 \leq i \leq k$,

$$\bar{Y}_i = \bar{Y}_i^{+} + \bar{Y}_i^{-}$$

where $n\bar{Y}_i^{+} \sim \mathrm{Bin}(n_i, 1-p)$, $n\bar{Y}_i^{-} \sim \mathrm{Bin}(n-n_i, p)$ are independent. We can then express

$$\begin{aligned}\mathbb{E}\left[\|\bar{Y}-\mathbb{E}\left[\bar{Y}\right]\|_\infty\right] &= \mathbb{E}\left[\max_{1\le i\le k}\left|\bar{Y}_i-\mathbb{E}\left[\bar{Y}_i\right]\right|\right]\\ &= \mathbb{E}\left[\max_{1\le i\le k}\left(\left|\bar{Y}_i^+-\mathbb{E}\left[\bar{Y}_i^+\right]\right|+\left|\bar{Y}_i^--\mathbb{E}\left[\bar{Y}_i^-\right]\right|\right)\right]\\ &\le \mathbb{E}\left[\max_{1\le i\le k}\left|\bar{Y}_i^+-\mathbb{E}\left[\bar{Y}_i^+\right]\right|\right]+\mathbb{E}\left[\max_{1\le i\le k}\left|\bar{Y}_i^--\mathbb{E}\left[\bar{Y}_i^-\right]\right|\right] \qquad (6)\end{aligned}$$

Now, instead of taking the maximum of k sums of Bernoullis with different parameters but same number of summands (n summands), we take the maximum of k sums of Bernoullis with the same parameter (i.e., Binomials) but different number of summands (at most n). This does not necessarily seem like an improvement, and still does not let us apply Theorem 16 to either of the two expectations. However, *if* we could argue that "adding summands to each Binomial" cannot decrease the expected maximum, then we would be in good shape: that is, we want to upper bound

$$\mathbb{E}\left[\max_{1\le i\le k}\left|\bar{Y}_i^+-\mathbb{E}\left[\bar{Y}_i^+\right]\right|\right]$$

by

$$\mathbb{E}\left[\max_{1\le i\le k}\left|Z_i^+-\mathbb{E}\left[Z_i^+\right]\right|\right]$$

where $nZ_i^+ \sim \mathrm{Bin}(n, 1-p)$ (instead of $\mathrm{Bin}(n_i, 1-p)$). Intuitively, this seems reasonable, as adding independent summands should make the Binomial more likely to deviate from its expectation. The next lemma makes this intuition rigorous:

▶ **Lemma 17.** *Fix $n_1,\dots,n_k,m_1,\dots,m_k \in \mathbb{N}$ and $p_1,\dots,p_k \in [0,1]$, with $n_i \le m_i$ for all i. Let $N_1,\dots,N_k$ and $M_1,\dots,M_k$ be (not necessarily independent) random variables with $N_i \sim \mathrm{Bin}(n_i,p_i)$ and $M_i \sim \mathrm{Bin}(m_i,p_i)$. Then*

$$\mathbb{E}\left[\max_{1\le i\le k}|N_i-\mathbb{E}[N_i]|\right] \le \mathbb{E}\left[\max_{1\le i\le k}|M_i-\mathbb{E}[M_i]|\right].$$

Proof. Set $\tilde{N}_i \coloneqq N_i-\mathbb{E}[N_i]$ and $\tilde{M}_i \coloneqq M_i-\mathbb{E}[M_i]$ for all $1\le i\le k$. We give a coupling of $\tilde{N}_i, \tilde{M}_i$ such that

$$\mathbb{E}\left[\,\tilde{M}_i \;\middle|\; \tilde{N}_i\,\right] = \tilde{N}_i\,.$$

Such a coupling can be obtained by setting $\tilde{M}_i = \tilde{N}_i+\Delta_i-\mathbb{E}[\Delta_i]$, where $\Delta_i \sim \mathrm{Bin}(m_i-n_i,p_i)$ is independent of $\tilde{N}_i$. Then it is easy to check that $\tilde{M}_i$ has the right distribution, since $N_i+\Delta_i \sim \mathrm{Bin}(m_i,p_i)$; and the the conditional expectation is indeed as claimed. Using this coupling, we obtain

$$\begin{aligned}\mathbb{E}\left[\max_{1\le i\le k}|\tilde{N}_i|\right] &= \mathbb{E}\left[\max_{1\le i\le k}\left|\mathbb{E}\left[\,\tilde{M}_i \;\middle|\; \tilde{N}_i\,\right]\right|\right]\\ &\le \mathbb{E}\left[\mathbb{E}\left[\max_{1\le i\le k}|\tilde{M}_i| \;\middle|\; \tilde{N}_i\right]\right] && \text{(Jensen's inequality)}\\ &= \mathbb{E}\left[\max_{1\le i\le k}|\tilde{M}_i|\right]\end{aligned}$$

establishing the lemma.[8] ◀

[8] More generally, via the existence of this coupling the argument shows that $(\tilde{N}_1,\dots,\tilde{N}_k) \preceq_{\mathrm{cx}} (\tilde{M}_1,\dots,\tilde{M}_k)$ (domination in the convex order), which in turn is equivalent to having $\mathbb{E}\left[\phi(\tilde{N}_1,\dots,\tilde{N}_k)\right] \le \mathbb{E}\left[\phi(\tilde{M}_1,\dots,\tilde{M}_k)\right]$ for every convex function ϕ.

Let $Z_1^+, \ldots, Z_k^+$ (resp. $Z_1^-, \ldots, Z_k^-$) be i.i.d. with $nZ_i^+ \sim \mathrm{Bin}(n, 1-p)$ random variables (resp. $nZ_i^- \sim \mathrm{Bin}(n,p)$). Invoking Lemma 17 with $m_1 = \cdots = m_k = n$ separately on the two expectations of Equation (6), we get

$$\mathbb{E}\left[\|\bar{Y} - \mathbb{E}\left[\bar{Y}\right]\|_\infty\right] \leq \mathbb{E}\left[\max_{1\leq i\leq k} \left|Z_i^+ - \mathbb{E}\left[Z_i^+\right]\right|\right] + \mathbb{E}\left[\max_{1\leq i\leq k} \left|Z_i^- - \mathbb{E}\left[Z_i^-\right]\right|\right]$$

Both of the terms in the RHS now fit the setting of Theorem 16. Moreover, since their parameters are p (for the first expectation) and $1-p$ (for the second), in both case the corresponding $\tilde{mu}_i = \min(p, 1-p) = p = \frac{1}{e^{\varepsilon/2}+1}$ is the same, and applying the theorem will give the same upper bound for both expectations. Thus, Theorem 16 yields

$$\mathbb{E}\left[\|\bar{Y} - \mathbb{E}\left[\bar{Y}\right]\|_\infty\right] \lesssim \sqrt{\frac{p\log(k+1)}{n}} + \frac{\log n}{n} \cdot \frac{\log(k+1)}{\log\frac{1}{p}}$$

Recalling the setting of p, along with the fact that $\mathbb{E}[\|\hat{q} - q\|_\infty] = \frac{e^{\varepsilon/2}+1}{e^{\varepsilon/2}-1}\mathbb{E}\left[\|\bar{Y} - \mathbb{E}\left[\bar{Y}\right]\|_\infty\right]$, finally gives

$$\mathbb{E}\left[\|\bar{Y} - \mathbb{E}\left[\bar{Y}\right]\|_\infty\right] \lesssim \sqrt{\frac{\log(k+1)}{n} \cdot \frac{e^{\varepsilon/2}+1}{(e^{\varepsilon/2}-1)^2}} + \frac{\log n}{n} \cdot \frac{e^{\varepsilon/2}+1}{e^{\varepsilon/2}-1} \cdot \frac{\log(k+1)}{\log(e^{\varepsilon/2}+1)} \tag{7}$$

To conclude, observe that the right hand side is $O\left(\sqrt{\frac{\log k}{n\varepsilon^2}} + \frac{\log k}{n\varepsilon} \cdot \log n\right)$ for $\varepsilon \leq 1$, and $O\left(\sqrt{\frac{\log k}{ne^{\varepsilon/2}}} + \frac{\log k}{n\varepsilon} \cdot \log n\right)$ for $\varepsilon \geq 1$. ◀

3.4 Upper bound for Projective Geometry Response

Projective geometry response (PGR), introduced by [24] achieves optimal rates for ℓ_2^2 error, communication, and near-optimal processing time. The protocol is based on the general template established in [4]: specifically, PGR relies on a set structure defined by *projective planes*, as detailed next. For a prime power d and $k = \frac{d^t-1}{d-1}$, the authors define a t-dimensional vector space $\mathbb{F}_d^t$, where each element $x \in [k]$ is represented by one of the canonical basis vectors. These basis vectors in turn each uniquely determine a projective plane $S(x)$, such that there are $s = |S(x)| = \frac{d^{t-1}-1}{d-1}$ "high probability" elements. Every one of these sets in turn has an intersection with every other set of size $c = |S(x) \cap S(x')| = \frac{d^{t-2}-1}{d-1}$. By choosing a prime power $d \approx e^\varepsilon + 1$, optimal error is achieved.

▶ **Theorem 18.** *Projective geometry response (PGR) [24] achieves the optimal rate for ℓ_∞ error. More specifically, the expected ℓ_∞ error of Projective Geometry Response for frequency estimation satisfies*

$$\begin{aligned}\mathbb{E}[\|\hat{q} - q\|_\infty] &\leq \sqrt{\frac{16(2e^\varepsilon+1)^2\log(k+1)}{e^\varepsilon(e^\varepsilon-1)^2 n}} + \frac{4(2e^\varepsilon+1)\log(k+1)}{(e^\varepsilon-1)\varepsilon n}\log n \\ &\in O\left(\sqrt{\frac{\log k}{ne^\varepsilon}} + \frac{\log k}{n\varepsilon}\log n\right)\end{aligned}$$

We will make use of the size of each subset s and the size of each intersection c as described above, as well as the probability of returning any element y from the output alphabet, given x as an input:

$$Q(Y = y \mid X = x) = \begin{cases} \frac{e^\varepsilon}{se^\varepsilon + k - s} & \text{if } y \in S(x) \\ \frac{s-c}{se^\varepsilon + k - s} & \text{otherwise.} \end{cases} \tag{8}$$

The estimate $\hat{q}$ of the frequency vector is then given by

$$\hat{q}_x \coloneqq \alpha \cdot \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{Y_i \in S(x)} + \beta, \qquad x \in [k] \tag{9}$$

where

$$\alpha = \frac{(e^\varepsilon - 1)s + k}{(e^\varepsilon - 1)(s - c)}, \qquad \beta = -\frac{(e^\varepsilon - 1)c + s}{(e^\varepsilon - 1)(s - c)} \tag{10}$$

so that $\hat{q}$ is an unbiased estimator of the true frequency vector q. Since a prime power can be found within a factor 2 of any number, we can choose d such that

$$e^\varepsilon + 1 \leq d \leq 2(e^\varepsilon + 1) \tag{11}$$

(While we could instead choose $d < e^\varepsilon + 1$ and set the inequality to be an undershoot by a factor two, an investigation in that direction led to worse constants and trickier analysis.) Recall that, for every integer ℓ,

$$d^\ell - 1 = (d-1)(1 + d + \ldots + d^{\ell-1}), \tag{12}$$

an identity we will rely on extensively in the rest of the section.

▶ Remark 19. Note that, as introduced above, we must have $k = \frac{d^t - 1}{d-1}$ for a prime power d satisfying (11). Other values of k must be rounded up, losing up to a factor $O(e^\varepsilon)$ in the domain size (i.e., working instead with a domain size $k' = O(e^\varepsilon k)$). We hereafter ignore this detail, which does not affect the final bound of Theorem 18 unless $\varepsilon \gg \log k$; and assume that k is of the form stated above. For the same reason, we additionally can assume $t \geq 3$.

Proof of Theorem 18. We start with two lemmas which will be useful in proving the optimal error rate.

▶ **Lemma 20.** *The common size s of every subset $S(x)$, $x \in [k]$, satisfies $s \geq e^\varepsilon + 2$.*

Proof. We can rewrite s as

$$s = \frac{d^{t-1} - 1}{d - 1} = \sum_{i=0}^{t-2} d^i$$

As $t \geq 3$, we have, by applying Equations (11) and (12)

$$s = \frac{d^{t-1} - 1}{d - 1} = \sum_{i=0}^{t-2} d^i \geq 1 + d \geq e^\varepsilon + 2. \qquad \blacktriangleleft$$

Now remembering that whatever expected error we compute will be multiplied by the normalising constant α, we would like to bound that in terms of ε.

▶ **Lemma 21.** *The value α defined in* (10) *satisfies $\alpha \leq 2 + \frac{2+d}{e^\varepsilon - 1} \leq 2 + \frac{4 + 2e^\varepsilon}{e^\varepsilon - 1}$.*

Proof. First note that

$$\alpha = \frac{(e^\varepsilon - 1)s + k}{(e^\varepsilon - 1)(s - c)} = \frac{s}{(s - c)} + \frac{k}{(e^\varepsilon - 1)(s - c)}$$

and note that by simple application of Equation (12) we get $s - c = d^{t-2}$. Applying the identity again we get:

$$\frac{s}{d^{t-2}} = \sum_{i=0}^{t-2} \frac{1}{d^{i-t+2}} = \frac{1}{d^{t-2}} + \frac{1}{d^{t-1}} + \ldots + 1 \leq 2 \tag{13}$$

where the final inequality comes from the fact that with $d \geq 2$ this is bounded by the geometric series. For k we have the same series, with the addition of a single d term.

$$\frac{k}{d^{t-2}} = \sum_{i=0}^{t-1} \frac{1}{d^{i-t+2}} = \frac{1}{d^{t-2}} + \frac{1}{d^{t-1}} + \ldots + 1 + d$$

The result is then at most $2 + d \leq 2 + 2(e^\varepsilon + 1)$, by applying Equation (11). ◀

We are now ready to apply Theorem 16. As we did for RAPPOR, we will break the expected maximum into the sum of error over two vectors of binomials $Z^+ = \mathrm{Bin}(n, p^+)$ and $Z^- = \mathrm{Bin}(n, p^-)$ where, recalling (8),

$$p^+ = \frac{e^\varepsilon}{se^\varepsilon + k - s}, \qquad p^- = \frac{s - c}{se^\varepsilon + k - s} \tag{14}$$

In this case we do not have that $p^+ = 1 - p^-$ so we will need to bound the following

$$\mathbb{E}\left[\|\hat{q} - q\|_\infty\right] \leq \alpha \left(\sqrt{\frac{\log(k+1)}{n}} \left(\sqrt{p^+} + \sqrt{p^-} \right) + \frac{\log(k+1)}{n} \log(n) \left(\frac{1}{\log(1/p^+)} + \frac{1}{\log(1/p^-)} \right) \right).$$

We briefly note that while RAPPOR has clear independence between coordinates, the result of Theorem 16 due to [17] does not require independence; therefore it applies to PGR, and is likely suitable for the analysis of other subset-based protocols.

First we will upper bound $\sqrt{p^+} + \sqrt{p^-}$:

▶ **Lemma 22.** *For p^+, p^- defined in (14), we have $p^+ + p^- \leq 2/e^\varepsilon$, and so $\sqrt{p^+} + \sqrt{p^-} \leq \sqrt{4/e^\varepsilon}$.*

Proof. First, p^+ is clearly less than $\frac{1}{s}$ when we remove $k - s$ from the denominator, and so is at most $1/(e^\varepsilon + 2) \leq 1/e^\varepsilon$ by Lemma 20. For the other term, notice that by Equation (13) we have $s/(s - c) \geq 1$, from which,

$$p^- \leq \frac{s - c}{se^\varepsilon} \leq \frac{1}{e^\varepsilon}$$

Overall, we get that $p^+ + p^- \leq 2/e^\varepsilon$. The conclusion follows from the AM-GM inequality, as $\sqrt{p^+} + \sqrt{p^-} \leq \sqrt{2(p^+ + p^-)}$. ◀

Next we bound $\log^{-1}(1/p^+) + \log^{-1}(1/p^-)$:

▶ **Lemma 23.** *We have $\frac{1}{\log(1/p^+)} + \frac{1}{\log(1/p^-)} \leq \frac{2}{\varepsilon}$.*

Proof. We will proceed in both cases by lower-bounding the denominators. First,

$$\log(1/p^-) = \log\left(\frac{se^\varepsilon + k - s}{s - c}\right) \geq \log\left(\frac{se^\varepsilon}{s - c}\right) \geq \log(e^\varepsilon) = \varepsilon.$$

Next,

$$\begin{aligned}\log(1/p^+) = \log\left(\frac{se^\varepsilon + k - s}{e^\varepsilon}\right) &\geq \log(s) \\ &\geq \log(e^\varepsilon + 2) && \text{(Lemma 20)} \\ &\geq \varepsilon\end{aligned}$$

As such adding the reciprocals of both terms give a bound of $2/\varepsilon$. ◀

The only step left to bound the ℓ_∞ error of PGR is to take the product of these bounds. Proceeding to do so, we arrive at:

$$\mathbb{E}[\|\hat{q} - q\|_\infty] \leq \sqrt{\frac{16(2e^\varepsilon + 1)^2 \log(k+1)}{e^\varepsilon(e^\varepsilon - 1)^2 n}} + \frac{4(2e^\varepsilon + 1)\log(k+1)}{(e^\varepsilon - 1)\varepsilon n}\log n$$

which in the low privacy regime gives:

$$\mathbb{E}[\|\hat{q} - q\|_\infty] \in O\left(\sqrt{\frac{\log k}{ne^\varepsilon}} + \frac{\log k}{n\varepsilon}\log n\right)$$

as claimed. ◀

4 Worst-case, information-theoretic lower bounds

We will follow the "chi-squared lower bound" framework of [2] to obtain our information-theoretic lower bounds against non-interactive locally private protocols:

▶ **Theorem 24.** *Fix any $\varepsilon > 0$. Any non-interactive (public- or private-coin) protocol Π for distribution estimation from n users must have minmax expected ℓ_∞ error*

$$\Omega\left(\max\left(\sqrt{\frac{\log k}{n(e^\varepsilon - 1)^2}}, \sqrt{\frac{\log k}{ne^\varepsilon}}, \frac{\log k}{n\varepsilon}\right)\right)$$

Proof. Suppose there exists a (non-interactive, public- or private-coin) ε-LDP protocol Π for n users which learns any $\mathbf{p}$ to expected ℓ_∞ error α when each user gets an independent sample from $\mathbf{p}$ as input:

$$\mathbb{E}[\|\mathbf{p} - \hat{\mathbf{p}}\|_\infty] \leq \alpha \tag{15}$$

where $\hat{\mathbf{p}}$ is the output of Π when run on $X_1, \ldots, X_n \sim \mathbf{p}$.

Now, consider the family $\mathcal{P}_\alpha = \{\mathbf{p}_z\}_{z\in[k]}$ of probability distributions over $[k]$, where, for $z \in [k]$, $\mathbf{p}_z$ is defined by

$$\mathbf{p}_z(x) = \frac{1 - 4\alpha}{k} + 4\alpha\mathbb{1}_{\{z=x\}}, \qquad x \in [k] \tag{16}$$

(that is, $\mathbf{p}_z$ is a mixture of the uniform distribution $\mathbf{u}_k$ and a point mass on z, with mixture coefficients $1 - 4\alpha$ and 4α). Consider the following process: first, we select Z from $[k]$ uniformly at random, then generate n i.i.d. samples $X_1, \ldots, X_n$ from $\mathbf{p}_Z$ and run Π on these samples, obtaining a hypothesis $\hat{\mathbf{p}}$. We finally set $\hat{Z}$ to be the element of the domain with the largest probability under $\hat{\mathbf{p}}$, i.e.,

$$\hat{Z} = \operatorname*{argmax}_{i\in[k]} \hat{\mathbf{p}}(i)$$

(breaking ties arbitrarily). The first claim is that doing so allows one to guess the correct value of Z with high probability: indeed, since the gap between the highest and second highest element of $\mathbf{p}_Z$ is 4α, we have

$$\Pr\Big[\,\hat{Z} = Z\,\Big] \geq \Pr[\,\|\mathbf{p} - \hat{\mathbf{p}}\|_\infty < 2\alpha\,] \geq 1 - \frac{\mathbb{E}[\|\mathbf{p}_Z - \hat{\mathbf{p}}\|_\infty]}{2\alpha} \geq \frac{1}{2} \tag{17}$$

the second-to-last inequality being Markov's. However, by Fano's inequality applied to the Markov chain $Z - \mathbf{p}_Z - X^n - (Y^n, R) - \hat{\mathbf{p}} - \hat{Z}$ (where X^n is the tuple of i.i.d. samples, and (Y^n, R) denotes the tuple of n messages, along with the public randomness, resulting from the protocol Π), we get, recalling that Z is chosen uniformly in $[k]$, that

$$\Pr\Big[\,\hat{Z} = Z\,\Big] \leq \frac{I(Z \wedge (Y^n, R)) + \log 2}{\log k} \tag{18}$$

and so, putting Equations (17) and (18) together, we get

$$I(Z \wedge (Y^n, R)) \geq \frac{1}{2}\log\frac{k}{4} = \Omega(\log k)\,. \tag{19}$$

This gives us the first ingredient: a lower bound on $I(Z \wedge (Y^n, R))$. For the second, we need to obtain an upper bound on this same mutual information as a function of ε, n, k, and α. To do so, observe first that, by the chain rule

$$I(Z \wedge (Y^n, R)) = I(Z \wedge Y^n \mid R) + I(Z \wedge R) = I(Z \wedge Y^n \mid R)$$

the second equality follows from the independence of the public randomness R from Z. This is convenient, as the messages $Y^n = (Y_1, \ldots, Y_n)$ are independent conditioned on R, and so we get

$$I(Z \wedge (Y^n, R)) \leq \sum_{j=1}^{n} I(Z \wedge Y_j \mid R) \leq n \cdot \max_{1 \leq j \leq n} I(Z \wedge Y_j \mid R) \tag{20}$$

Consider any user j, using locally private randomizer $Q = Q_{j,R} \colon [k] \to \mathcal{Y}$ (for notational simplicity, we drop afterwards the dependence on j and R). Let $\bar{\mathbf{p}} = \mathbb{E}_Z[\mathbf{p}_Z]$ denote the average input distribution (over Z), i.e., the uniform mixture of all $\mathbf{p}_z$'s. Then we can rewrite the mutual information as

$$I(Z \wedge Y_j \mid R) = \mathbb{E}_Z\big[\mathrm{KL}\big(Q^{\mathbf{p}_Z} \,||\, Q^{\bar{\mathbf{p}}}\big)\big] \tag{21}$$

where, for a given input distribution $\mathbf{p}$, $Q^{\mathbf{p}}(\cdot) = \mathbb{E}_{X \sim \mathbf{p}}[Q(\cdot \mid X)]$ denotes the output distribution (over $\mathcal{Y}$) induced by the randomizer Q on input $X \sim \mathbf{p}$.

First lower bound (good for small ε). We then proceed by upperbounding the KL divergence by the χ^2 one and unrolling the latter's definition, getting

$$\begin{aligned} I(Z \wedge Y_j \mid R) \leq \mathbb{E}_Z\big[\chi^2\big(Q^{\mathbf{p}_Z} \,||\, Q^{\bar{\mathbf{p}}}\big)\big] &= \mathbb{E}_Z\left[\sum_{y \in \mathcal{Y}} \frac{(\mathbb{E}_{X \sim \mathbf{p}_Z}[Q(y \mid X)] - \mathbb{E}_{X \sim \bar{\mathbf{p}}}[Q(y \mid X)])^2}{\mathbb{E}_{X \sim \bar{\mathbf{p}}}[Q(y \mid X)]}\right] \\ &= \mathbb{E}_Z\left[\sum_{y \in \mathcal{Y}} \frac{\Big(\sum_{x \in [k]} Q(y \mid x)(\mathbf{p}_Z(x) - \bar{\mathbf{p}}(x))\Big)^2}{\sum_{x \in [k]} Q(y \mid x)\bar{\mathbf{p}}(x)}\right] \end{aligned}$$

Note, observing that, for our choice of $\mathcal{P}_\alpha$, $\bar{\mathbf{p}}$ is simply the uniform distribution over $[k]$, and that for all $x \in [k]$ we then have $\mathbf{p}_Z(x) - \bar{\mathbf{p}}(x) = 4\alpha\big(\mathbb{1}_{\{Z=x\}} - \frac{1}{k}\big)$. Then, letting $\Phi(x) = e_x - \frac{1}{k}\mathbf{1}_k \in \mathbb{R}^k$ for $x \in [k]$, we get

$$\begin{aligned} I(Z \wedge Y_j \mid R) &\le 16\alpha^2 k^2 \cdot \mathbb{E}_Z\left[\sum_{y\in\mathcal{Y}} \frac{\big(\mathbb{E}_{X\sim\mathbf{u}}\big[Q(y \mid X)\big(\mathbb{1}_{\{Z=X\}} - \frac{1}{k}\big)\big]\big)^2}{\mathbb{E}_{X\sim\mathbf{u}}[Q(y \mid X)]}\right] \\ &= 16\alpha^2 k^2 \cdot \sum_{y\in\mathcal{Y}} \frac{\mathbb{E}_Z\Big[\big(\mathbb{E}_{X\sim\mathbf{u}}\big[Q(y \mid X)\big(\mathbb{1}_{\{Z=X\}} - \frac{1}{k}\big)\big]\big)^2\Big]}{\mathbb{E}_{X\sim\mathbf{u}}[Q(y \mid X)]} \\ &= 16\alpha^2 k \cdot \sum_{y\in\mathcal{Y}} \frac{\sum_{i=1}^k \big(\mathbb{E}_{X\sim\mathbf{u}}\big[Q(y \mid X)\big(\mathbb{1}_{\{i=X\}} - \frac{1}{k}\big)\big]\big)^2}{\mathbb{E}_{X\sim\mathbf{u}}[Q(y \mid X)]} \\ &= 16\alpha^2 k \cdot \sum_{y\in\mathcal{Y}} \frac{\sum_{i=1}^k \mathbb{E}_{X\sim\mathbf{u}}[Q(y \mid X)\Phi(X)_i]^2}{\mathbb{E}_{X\sim\mathbf{u}}[Q(y \mid X)]} \end{aligned} \tag{22}$$

Note that, for any fixed $1 \le i < j \le k$, we have

$$\mathbb{E}_{X\sim\mathbf{u}}[\Phi_i(X)] = 0, \quad \mathbb{E}_{X\sim\mathbf{u}}\big[\Phi_i(X)^2\big] = \frac{1}{k}\left(1 - \frac{1}{k}\right), \quad \mathbb{E}_{X\sim\mathbf{u}}[|\Phi_i(X)|] = \frac{2}{k}\left(1 - \frac{1}{k}\right),$$

For simplicity, we write $\mathbb{E}_X[\cdot]$ for $\mathbb{E}_{X\sim\mathbf{u}}[\cdot]$. We will use the fact that, Q being ε-LDP, we have

$$|Q(y \mid x) - \mathbb{E}_X[Q(y \mid X)]| \le (e^\varepsilon - 1)\mathbb{E}_X[Q(y \mid X)] \tag{23}$$

for every $x \in [k]$ and $y \in \mathcal{Y}$. With this in hand, starting from (22), we can write

$$I(Z \wedge Y_j \mid R) \le 16\alpha^2 k \sum_{y\in\mathcal{Y}} \frac{\sum_{i=1}^k \mathbb{E}_X[Q(y \mid X)\Phi(X)_i]^2}{\mathbb{E}_X[Q(y \mid X)]} \le 64\alpha^2(e^\varepsilon - 1)^2\,.$$

where the derivation relies on the fact that $\mathbb{E}_X[\Phi_i(X)] = 0$, (23), and $\mathbb{E}_{X\sim\mathbf{u}}[|\Phi_i(X)|] \le \frac{2}{k}$ (details can be found in the full version). Using this last bound along with Equations (19) and (20), we get that

$$\frac{1}{2}\log\frac{k}{4} \le I(Z \wedge (Y^n, R)) \le 64\alpha^2 n(e^\varepsilon - 1)^2\,, \tag{24}$$

i.e.,

$$\alpha \ge \frac{1}{8\sqrt{2}}\sqrt{\frac{\log\frac{k}{4}}{n(e^\varepsilon - 1)^2}}, \tag{25}$$

showing the $\Omega\left(\sqrt{\frac{\log k}{n\varepsilon^2}}\right)$ lower bound for small ε.

The second and third lower bounds can be established in an analogous fashion: the details can be found in the full version. ◀

5 Amplification by shuffling

As mentioned in the Introduction, one of the key motivations for studying locally private histogram estimation in the low privacy (high-ε) regime is the implication for histogram estimation in the *shuffle* model of privacy (see, e.g., [27, 26, 7, 16]), in light of the "plug-and-play" amplification-by-shuffling results allowing to "translate" the former into the latter. Specifically, we will use the following result of Feldman, McMillan, and Talwar:

▶ **Theorem 25** ([23, Theorem 3.1]). *For any domain $\mathcal{X}$, let $R\colon \mathcal{X} \to \mathcal{Y}$ be an ε_L-DP local randomiser; and let S be the algorithm that given a tuple of n messages $\vec{y} \in \mathcal{Y}^n$, samples a uniform random permutation π over $[n]$ and outputs $(\vec{y}_{\pi(1)}, \ldots, \vec{y}_{\pi(n)})$. Then for any $\delta \in (0,1]$ such that $\varepsilon_L \leq \log \frac{n}{16\log(2/\delta)}$, $S \circ R^n$ is (ε,δ)-DP, where*

$$\varepsilon \leq \log\left(1 + 8\frac{e^{\varepsilon_L}-1}{e^{\varepsilon_L}+1}\left(\sqrt{\frac{e^{\varepsilon_L}\log(4/\delta)}{n}} + \frac{e^{\varepsilon_L}}{n}\right)\right).$$

In particular, if $\varepsilon_L \geq 1$ then $\varepsilon = O\Big(\sqrt{e^{\varepsilon_L}\log(1/\delta)/n}\Big)$, and if $\varepsilon_L < 1$ then $\varepsilon = O\Big(\varepsilon_L\sqrt{\log(1/\delta)/n}\Big)$.

This implies the following:

▶ **Lemma 26** (Amplification by shuffling). *Fix any $\delta \in (0,1]$, $\varepsilon \in (0,1]$, and n such that $\varepsilon > 16\sqrt{\log(4/\delta)/n}$. Then, for*

$$\varepsilon_L \coloneqq \log\frac{\varepsilon^2 n}{256\log(4/\delta)} = \Theta\left(\log\frac{\varepsilon^2 n}{\log(1/\delta)}\right),$$

shuffling the messages of n users using the same ε_L-LDP randomizer satisfies (robust) (ε,δ)-shuffle differential privacy.

Proof. Note that for ε, δ as in the statement and ε_L as defined, we have $0 < \varepsilon_L \leq \log\frac{n}{16\log(2/\delta)}$. Applying Theorem 25, we get (ε', δ) privacy for

$$\varepsilon' = \log\Big(1 + 8\underbrace{\frac{e^{\varepsilon_L}-1}{e^{\varepsilon_L}+1}}_{\leq 1}\Big(\frac{\varepsilon}{16} + \underbrace{\frac{\varepsilon^2}{256\log(4/\delta)}}_{\leq \varepsilon/16}\Big)\Big) \leq \varepsilon$$

which proves the statement. ◀

▶ **Theorem 27.** *For $n = \Omega\Big(\frac{\log(1/\delta)}{\varepsilon^2}\Big)$ and $\varepsilon \in (0,1]$, Shuffled Projective Geometry Response achieves maximum error $O\left(\frac{\sqrt{\log(k)\log(1/\delta)}}{n\varepsilon}\right)$, with $O(\log k)$ bits of communication (and one single message) per user.*

Proof. For $n \geq \frac{500}{\varepsilon^2}\log\frac{4}{\delta}$, the restriction on ε from Lemma 26 holds, and, setting ε_L as in the lemma, we also have $\varepsilon_L = \Omega(1)$. We then invoke the bound of Theorem 18, focusing on first part of the bound, which dominates when $\frac{\log k}{n} \cdot \log^2 n \leq \frac{\varepsilon_L^2}{e^{\varepsilon_L}}$. This leads to an upper bound on the error of the order

$$\sqrt{\frac{\log k}{ne^{\varepsilon_L}}} = \sqrt{\frac{\log(k)}{n\varepsilon^2\frac{n}{\log(1/\delta)}}} = \frac{\sqrt{\log(k)\log(1/\delta)}}{n\varepsilon}$$

as desired. It only remains to argue that the first term of the bound did, indeed, dominate the error. As mentioned above, this is the case whenever

$$\frac{e^{\varepsilon_L}}{\varepsilon_L^2} \cdot \log k \ll \frac{n}{\log^2 n}$$

for which a weaker, sufficient condition is $\frac{e^{\varepsilon_L}}{\varepsilon_L^2} \ll \frac{n}{\log^2 n}$, that is, $n \gg e^{\varepsilon_L}$. But this follows from our setting of ε_L, such that $e^{\varepsilon_L} = \underbrace{\frac{\varepsilon^2}{256\log(4/\delta)}}_{\ll 1} \cdot n$. ◀

Comparing this result with the summary of local, shuffle, and central histogram error bounds available in [16, Table 1] shows that with shuffled PGR achieves the best error of any protocol which sends a constant number of messages.

More specifically, focusing on 3 representative known protocols: the only known one-message-per-user protocol, due to [16], achieves much worse error,[9] and requires k bits of communication per user; while the protocol of [26], which achieves the same error as Theorem 27, uses $k^{\Omega(1)}$ $(\log k)$-bit messages per user. Finally, a protocol of [6] does achieve better error, but at the cost of performing $k+1$ rounds, each with $\log k$ communication per user. Thus, our bounds demonstrates that shuffled PGR achieve state-of-the-art ℓ_∞–error with only *one* message of $\log k$ bits.

6 Discussion and future work

The logarithmic factor in the upper bound. We prove our upper bounds using the result of [17], which is where the $\log n$ term appears. The authors conjectured that this term could be removed in general, but follow–up work [10] demonstrated by counter–example that this is not the case. They do, however, show that there is a distribution–dependent interpolation between the term being $T(n)/n$ and $T(n)\log n/n$. This logarithmic factor has implications for understanding which error regime is dominating, given the parameters of the algorithm (ε, k, n). While it is possible to imagine that a careful application of the tools in [10] could resolve this matter in general or for a specific use–case, in the meantime we emphasise that empirical analysis is still crucial.

Tighter analysis of other protocols. While we analyse RAPPOR which admits a simple analysis due to the independence of coordinates, and Projective Geometry Response which represents the state of the art in low-communication LDP protocols, we believe the tools introduced in this paper are applicable in a very general way to most LDP protocols. Of particular interest would be Subset Selection, which has optimal mutual information between inputs and outputs of the local randomiser, and the two Hadamard Response protocols that we include in our empirical analysis.

Histograms in the shuffle model. While our shuffle DP result (Theorem 27) yields better error, communication, and number of messages, it does have a limitation on the parameter range, namely $n = \Omega(\log(1/\delta)/\varepsilon^2)$. It would be interesting to weaken this requirement to match the central DP one (where the dependence on ε is only linear). Moreover, one could hope to further improve the resulting error to the central DP bound (i.e., a $\min(\log k, \log(1/\delta))$ dependence instead of $\sqrt{\log k \log(1/\delta)}$), or, alternatively, prove a matching lower bound separating the two models.

Broader applicability of lower bound techniques. While we prove lower bounds for the specific case of ℓ_∞ frequency estimation, the tools used are extremely general and we imagine that their application could provide new lower bounds for a variety of problems in the LDP setting, especially with the low–privacy regime in mind.

[9] And has the same restriction $n = \Omega(\log(1/\delta)/\varepsilon^2)$ on the parameters.

References

1 Jayadev Acharya, Clément L. Canonne, Ziteng Sun, and Himanshu Tyagi. Unified lower bounds for interactive high-dimensional estimation under information constraints. In *NeurIPS*, 2023.

2 Jayadev Acharya, Clément L. Canonne, and Himanshu Tyagi. Inference under information constraints I: Lower bounds from chi-square contraction. *Institute of Electrical and Electronics Engineers*, 66(12):7835–7855, 2020. doi:10.1109/TIT.2020.3028440.

3 Jayadev Acharya and Ziteng Sun. Communication complexity in locally private distribution estimation and heavy hitters. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 51–60, Long Beach, California, USA, 2019. PMLR. URL: http://proceedings.mlr.press/v97/acharya19c.html.

4 Jayadev Acharya, Ziteng Sun, and Huanyu Zhang. Hadamard response: Estimating distributions privately, efficiently, and with little communication. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 1120–1129. PMLR, 2019. URL: http://proceedings.mlr.press/v89/acharya19a.html.

5 Apple Privacy Team. Learning with privacy at scale, 2017. URL: https://machinelearning.apple.com/research/learning-with-privacy-at-scale.

6 Victor Balcer and Albert Cheu. Separating Local & Shuffled Differential Privacy via Histograms, April 2020. arXiv:1911.06879, doi:10.48550/arXiv.1911.06879.

7 Victor Balcer and Albert Cheu. Separating local & shuffled differential privacy via histograms. In *ITC*, volume 163 of *LIPIcs*, pages 1:1–1:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.ITC.2020.1.

8 Raef Bassily and Adam Smith. Local, Private, Efficient Protocols for Succinct Histograms. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, pages 127–135, June 2015. doi:10.1145/2746539.2746632.

9 Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnés, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. *CoRR*, abs/1710.00901, 2017. arXiv:1710.00901.

10 Moïse Blanchard and Vaclav Voracek. Tight bounds for local glivenko-cantelli. In Claire Vernade and Daniel Hsu, editors, *Proceedings of The 35th International Conference on Algorithmic Learning Theory*, volume 237 of *Proceedings of Machine Learning Research*, pages 179–220. PMLR, February 2024. URL: https://proceedings.mlr.press/v237/blanchard24a.html.

11 Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. OUP Oxford, 2013.

12 V. V. Buldygin and K. K. Moskvichova. The sub-Gaussian norm of a binary random variable. *Theory of Probability and Mathematical Statistics*, 86:33–49, August 2013. doi:10.1090/S0094-9000-2013-00887-4.

13 Clément L. Canonne. A short note on learning discrete distributions, 2020. arXiv:2002.11457.

14 Wei-Ning Chen, Peter Kairouz, and Ayfer Özgür. Breaking the communication-privacy-accuracy trilemma. *IEEE Trans. Inf. Theory*, 69(2):1261–1281, 2023. doi:10.1109/TIT.2022.3218772.

15 Albert Cheu, Adam D. Smith, Jonathan R. Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *EUROCRYPT (1)*, volume 11476 of *Lecture Notes in Computer Science*, pages 375–403. Springer, 2019. doi:10.1007/978-3-030-17653-2_13.

16 Albert Cheu and Maxim Zhilyaev. Differentially private histograms in the shuffle model from fake users. In *SP*, pages 440–457. IEEE, 2022. doi:10.1109/SP46214.2022.9833614.

17 Doron Cohen and Aryeh Kontorovich. Local glivenko-cantelli. In *COLT*, volume 195 of *Proceedings of Machine Learning Research*, page 715. PMLR, 2023. URL: https://proceedings.mlr.press/v195/cohen23a.html.

18 Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. Privacy at scale: Local differential privacy in practice. In *SIGMOD Conference*, pages 1655–1658. ACM, 2018. doi:10.1145/3183713.3197390.

19 John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Local privacy and statistical minimax rates. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013*, pages 429–438. IEEE Computer Society, 2013. doi:10.1109/FOCS.2013.53.

20 John C. Duchi and Ryan Rogers. Lower bounds for locally private estimation via communication complexity. In *COLT*, volume 99 of *Proceedings of Machine Learning Research*, pages 1161–1191. PMLR, 2019. URL: http://proceedings.mlr.press/v99/duchi19a.html.

21 Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Shuang Song, Kunal Talwar, and Abhradeep Thakurta. Encode, Shuffle, Analyze Privacy Revisited: Formalizations and Empirical Evaluation. *arXiv:2001.03618 [cs]*, January 2020. arXiv:2001.03618.

22 Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM Conference on Computer and Communications Security*, CCS '14, pages 1054–1067, New York, NY, USA, 2014. ACM. doi:10.1145/2660267.2660348.

23 Vitaly Feldman, Audra McMillan, and Kunal Talwar. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 954–964. IEEE, 2021. doi:10.1109/FOCS52979.2021.00096.

24 Vitaly Feldman, Jelani Nelson, Huy Nguyen, and Kunal Talwar. Private frequency estimation via projective geometry. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 6418–6433. PMLR, July 2022. URL: https://proceedings.mlr.press/v162/feldman22a.html.

25 Vitaly Feldman and Kunal Talwar. Lossless compression of efficient private local randomizers. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3208–3219. PMLR, July 2021. URL: https://proceedings.mlr.press/v139/feldman21a.html.

26 Badih Ghazi, Noah Golowich, Ravi Kumar, Rasmus Pagh, and Ameya Velingker. On the power of multiple anonymous messages: Frequency estimation and selection in the shuffle model of differential privacy. In *EUROCRYPT (3)*, volume 12698 of *Lecture Notes in Computer Science*, pages 463–488. Springer, 2021. doi:10.1007/978-3-030-77883-5_16.

27 Badih Ghazi, Ravi Kumar, Pasin Manurangsi, and Rasmus Pagh. Private counting from anonymous messages: Near-optimal accuracy with vanishing communication overhead. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 3505–3514. PMLR, 2020. URL: http://proceedings.mlr.press/v119/ghazi20a.html.

28 Justin Hsu, Sanjeev Khanna, and Aaron Roth. Distributed private heavy hitters. In *ICALP (1)*, volume 7391 of *Lecture Notes in Computer Science*, pages 461–472. Springer, 2012. doi:10.1007/978-3-642-31594-7_39.

29 Ziyue Huang, Yuan Qiu, Ke Yi, and Graham Cormode. Frequency estimation under multiparty differential privacy: One-shot and streaming. *Proc. VLDB Endow.*, 15(10):2058–2070, 2022. doi:10.14778/3547305.3547312.

30 Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Extremal mechanisms for local differential privacy. *Journal of Machine Learning Research*, 17(17):1–51, 2016. URL: http://jmlr.org/papers/v17/15-135.html.

31 Sudeep Kamath, Alon Orlitsky, Dheeraj Pichapati, and Ananda Theertha Suresh. On Learning Distributions from their Samples. In *Proceedings of The 28th Conference on Learning Theory*, pages 1066–1100. PMLR, June 2015. URL: http://proceedings.mlr.press/v40/Kamath15.html.

32 Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019.

33 Shaowei Wang, Liusheng Huang, Pengzhan Wang, Yiwen Nie, Hongli Xu, Wei Yang, Xiang-Yang Li, and Chunming Qiao. Mutual information optimally local private discrete distribution estimation. *ArXiV*, abs/1607.08025, 2016. arXiv:1607.08025.

34 Min Ye and Alexander Barg. Optimal schemes for discrete distribution estimation under locally differential privacy. *Institute of Electrical and Electronics Engineers*, 64(8):5662–5676, 2018. doi:10.1109/TIT.2018.2809790.

Settling the Complexity of Testing Grainedness of Distributions, and Application to Uniformity Testing in the Huge Object Model

Clément L. Canonne
School of Computer Science, University of Sydney, Australia

Sayantan Sen
Centre for Quantum Technologies, National University of Singapore, Singapore

Joy Qiping Yang
School of Computer Science, University of Sydney, Australia

Abstract

In this work, we study the problem of testing *m-grainedness* of probability distributions over an n-element universe $\mathcal{U}$, or, equivalently, of whether a probability distribution is induced by a multiset $S \subseteq \mathcal{U}$ of size $|S| = m$. Recently, Goldreich and Ron (Computational Complexity, 2023) proved that $\Omega(n^c)$ samples are necessary for testing this property, for any $c < 1$ and $m = \Theta(n)$. They also conjectured that $\Omega(\frac{m}{\log m})$ samples are necessary for testing this property when $m = \Theta(n)$. In this work, we positively settle this conjecture.

Using a known connection to the Distribution over Huge objects (DoHo) model introduced by Goldreich and Ron (TheoretiCS, 2023), we leverage our results to provide improved bounds for uniformity testing in the DoHo model.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Distribution testing, Uniformity testing, Huge Object Model, Lower bounds

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.26

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2024/196` [12]

Funding *Clément L. Canonne*: Supported by an ARC DECRA (DE230101329).
Sayantan Sen: Supported by the National Research Foundation, Singapore and A*STAR under its Quantum Engineering Programme NRF2021-QEP2-02-P05.
Joy Qiping Yang: Supported by a JD Technology Research Scholarship in Artificial intelligence.

Acknowledgements We would like to thank the anonymous reviewers of ITCS 2025 for their suggestions which improved the presentation of the paper. SS would like to thank Clément Canonne and the Theory CS group at the University of Sydney for the warm hospitality during his academic visit, where this work was initiated.

1 Introduction

The field of distribution testing [7, 6] is concerned with providing statistically accurate information about large datasets or their underlying probability distributions, given very scarce data (sample size). Drawing insights from property testing [20, 27], distribution testing lies at the intersection of theoretical computer science, statistics, and learning theory; and has received significant attention over the past two decades, with many algorithms, insights, and new theoretical access models to the data being proposed and analyzed. We refer the reader to recent surveys [8, 4, 9] and textbook [19, Chapter 11] for more on distribution testing and property testing.


© Clément L. Canonne, Sayantan Sen, and Joy Qiping Yang;
licensed under Creative Commons License CC-BY 4.0



In the most standard access model, the algorithm accesses the underlying unknown probability distribution D (usually assumed to be over a known discrete domain of size n) by obtaining independent, identically distributed (i.i.d.) samples from it. The goal of the algorithm is then, given a fixed property $\Pi \subseteq \Delta_n$ (a subset of the n-dimensional probability simplex Δ_n) and a distance parameter $\varepsilon \in (0, 1]$, as follows:

- If $D \in \Pi$, the algorithm must output accept with probability at least $2/3$;
- If $\mathsf{d_{TV}}(D, Q) > \varepsilon$ for all $Q \in \Pi$, the algorithm must output reject with probability at least $2/3$;

where $\mathsf{d_{TV}}$ denotes the *total variation distance* between distributions, and the probability is taken over the choice of the samples and the internal randomness of the algorithm. This is defined as the *ε-testing* of the property Π. The minimum number of samples $s = s(n, \varepsilon)$ necessary to achieve this task in the worst case (over all possible distributions D) is the *sample complexity* of testing Π. That is, the testing task requires the algorithm, given as few samples as possible, to distinguish with high probability between distributions which have the property, and those which are "far" from having it.

Many other access models have been introduced, providing additional types of queries to the algorithm, or changing the distance metric, or both (see [8] for an overview): among them is the Distribution over Huge Objects (DoHO) model, recently introduced by Goldreich and Ron [23] to capture settings where full access to the data sampled is itself costly or impractical, due to the size of these objects. In the simplest version of this model, the distribution D is defined over the n-dimensional hypercube $\{0, 1\}^n$, where n is assumed to be very large; given a sample $x \sim D$, the algorithm must then choose which bits of x to observe, paying a unit cost for every such query made. The distance metric to quantify "farness" between distributions also differs from that of the standard formulation, and instead is taken to be the Earthmover distance (EMD) between probability distributions, with underlying metric chosen to be the (relative) Hamming distance between n-bit strings. (For the formal definition of this model, and the relation to the standard sampling model, see Section 1.3.)

Many different properties of distributions have been studied, some of them quite extensively: among those, *uniformity* ($\Pi = \{U_\Omega\}$, consisting of the single uniform distribution over the whole domain Ω), *generalized uniformity* [5] ($\Pi_U = \{U_S : S \subseteq \Omega\}$, consisting of all distributions uniform over *some* subset of the domain), and *parameterized uniformity* ($\Pi_m = \{U_S : S \subseteq \Omega, |S| = m\}$, consisting of all distributions uniform over *some* size-m subset of the domain) [23], and m-grainedness (denoted Π°_m, which we will define shortly) [18, 22] are the most relevant to this work.

In this paper, we will focus on two distribution testing tasks, intimately related:

- Testing m-grainedness of distributions in the standard sampling model (property Π°_m); and
- Testing parameterized uniformity in the DoHO model.

Grainedness of distributions

A probability distribution $D \in \Delta_n$ over a discrete set Ω of n elements is said to be *m-grained*, for a given parameter m, if all its probabilities are integer multiples of $1/m$. That is,

$$mD(x) \in \mathbb{N}^{\geq 0}, \qquad x \in \Omega\,.$$

Such distributions naturally arise due to quantization (e.g., binning of continuous or discrete distributions), or when sampling from datasets: that is, if $S \subseteq \Omega$ is a multiset of size m, uniformly sampling from S with replacement yields an m-grained distribution D_S over Ω.

Throughout this work, we will assume $\varepsilon \in (0,1)$ is a constant. Thus, $m < n/\varepsilon = O(n)$. Previous works fixed $m = \Theta(n)$ but these results can also be extended for $m = O(n)$, and therefore our results are stated in terms of m, instead of n.

Recent work of Goldreich and Ron [22] showed that $\Omega(m^c)$ samples are necessary for testing this property, for any fixed $c < 1$. A sample complexity upper bound of $O(\frac{m}{\log m})$ also follows from previous work of Valiant and Valiant [29], which led [22] to conjecture a lower bound of $\Omega(\frac{m}{\log m})$ samples. Our main contribution is to resolve this conjecture, showing that, indeed, $\Theta(\frac{m}{\log m})$ samples are necessary and sufficient for m-grainedness testing:

▶ **Theorem 1.1.** *Let n be a sufficiently large integer, $m = O(n)$, and let $\varepsilon \in (0,1)$ be a sufficiently small constant. Then, ε-testing m-grainedness of distributions over $[n]$ requires $\Omega(\frac{m}{\log m})$ samples.*

We note that the restriction $m = O(n)$ is necessary, as if $m = n/\alpha$ for some $\alpha < 1$, *every* distribution is α-close to being m-grained. Recall that we assume ε is a small constant. Moreover, the problem of testing m-grainedness becomes trivial when $m \geq n/\varepsilon$ as in that case every distribution is ε-close to being m-grained. Along the way, we also present as a warmup a new proof of the lower bound of $\Omega(n^c)$ samples for m-grainedness testing:

▶ **Theorem 1.2.** *Let n be a sufficiently large integer, $m = O(n)$, and let $\varepsilon \in (0,1)$ be a sufficiently small constant. Then, for any fixed constant $c \in (0,1)$, ε-testing m-grainedness of distributions over $[n]$ requires $\Omega(m^c)$ samples.*

While this is a strictly weaker result than our main lower bound, our proof strategy differs significantly from that of [22], and may be of independent interest.

Parameterized uniformity testing in the DoHO model

In a separate work [23], Goldreich and Ron studied the problem of parameterized uniformity testing in the Huge Object model. Among other results, they established a connection between m-grainedness testing in the *standard* model and testing Π_m in the DoHO model:

▶ **Theorem 1.3** ([23, Theorem 2.13])**.** *Assume that, for constant $\varepsilon \in (0,1)$, ε-testing m-grainedness in the standard model has sample complexity $\Omega(\frac{m}{\log m})$. Then, for every $1 \leq m \leq n$ and constant $\varepsilon' \in (0, \frac{\varepsilon}{2})$, ε'-testing Π_m for distribution over $\{0,1\}^n$ in the DoHO model has* query *complexity $\Omega(\frac{m}{\log m})$.*

Our lower bound for m-grainedness immediately implies that this result holds unconditionally. In the same paper, the authors provided an algorithm for testing Π_m using $\widetilde{O}(m)$ samples and queries (for constant $\varepsilon = \Omega(1)$), which together with the above – now unconditional – lower bound settles the complexity of testing Π_m in the DoHO model, up to logarithmic factors, *for $m \leq n$*. One may be tempted to assume the same query complexity lower bound holds in all parameter regimes: perhaps surprisingly, our next result is a new and simple algorithm for testing Π_m in the DoHO model which takes $O(m^{2/3})$ samples and performs $O(m^{2/3}n)$ queries:

▶ **Theorem 1.4.** *There is an algorithm which, given an integer $m \in \mathbb{N}$ and constant $\varepsilon \in (0,1)$, ε-tests the property Π_m of distributions over $\{0,1\}^n$ in the DoHO model, taking $\mathcal{O}(m^{2/3})$ samples and performing $\mathcal{O}(m^{2/3} \cdot n)$ queries. Moreover, any ε-tester for Π_m in the DoHO model must take $\Omega(m^{2/3})$ samples.*

Notably, this improves the upper bound of Goldreich and Ron whenever $n^3 \leq m \leq 2^n$, and rules out any $\tilde{\Omega}(m)$ query complexity lower bound for the whole range of m.

1.1 Overview of our techniques

Lower bound for m-grainedness testing

Our lower bound approach follows Le Cam's two-point method: we will design two distributions of distributions D_{yes} and D_{no} such that the distributions in D_{yes} are m-grained and the distributions in D_{no} are "far" from being m-grained in variation distance. The name of the game is then to prove that it is information-theoretically impossible to distinguish between D_{yes} and D_{no}.

To achieve this, we will rely on the *moment-matching* technique, particularly suited to properties like m-grained (which are symmetric, i.e., invariant to relabelling of the domain elements). Broadly, if two probability distributions D_{yes} and D_{no} have sufficiently similar moments $\|D_{\mathsf{yes}}\|_t^t \approx \|D_{\mathsf{no}}\|_t^t$, for say $1 \leq t \leq K$, then it is hard to distinguish (a permutation of) D_{yes} from (a permutation of) D_{no} by using $o(n^{1-1/K}/K)$ samples. That is, the more moments one can match, the more samples one needs to tell two distributions apart. Note that setting $K = \Theta(\log n)$ would then lead to an $\Omega(\frac{n}{\log n})$ lower bound (and that, as remarked later, in our case we can assume without loss of generality, for the sake of the lower bound argument, that $m = \Theta(n)$.) This approach has been used in the literature extensively [26, 29, 28, 30, 31, 32, 10], and has proven to be very successful in obtaining tight lower bounds for a range of symmetric properties. We will rely on the formulation of the technique described by [32], which maps the problem of matching moments of two full probability distributions (an n-dimensional object) to that of matching moments of two single-dimensional random variables U and V: these two random variables will then be used to generate the probability distributions: intuitively (and as described below), "n independent draws from U will give $D_{\mathsf{yes}}(1), \ldots, D_{\mathsf{yes}}(n)$" (and similarly for D_{no} and V).

Thus, a crucial ingredient in our lower bound is the construction of a pair of discrete random variables U and V whose first logarithmically many moments are identical. Using n independent copies of U (resp. V), we will then define a random measure over $[n]$, which corresponds to a "yes"-instance D_{yes} (resp., a "no"-instance D_{no}). Thus, another requirement on our construction of U and V is that the first one should yield an m-grained distribution, while the second should correspond to a distribution far from being m-grained. We will formalize this in Lemma 2.2.

To build these two random variables U and V, we will, as previously done in the literature, rely on the properties of the Chebyshev polynomial T_d of degree $d = O(\log n)$, defining U and V to be supported on disjoint subsets of the roots of a suitable polynomial $P(x)$, where the probability mass assigned to a given root r is proportional to $1/|P'(r)|$. The idea is that some of the roots of P, those corresponding to U, will be multiples of $1/m$ (leading to m-grained probabilities) while the others, corresponding to V, will be odd multiples of $1/(2m)$ (leading to "far-from-m-grained" probabilities): for instance, we would take

$$P(x) = x\left(x - \frac{1}{2m}\right)\left(1 - \frac{1}{m}\right)T_d(c \cdot x)$$

for some scaling constant $c > 0$. The remaining roots will be that of the Chebyshev polynomial T_d, which are there to ensure that we can match sufficiently many moments of U and V.

However, this approach, which underlies most of previous work, cannot be directly used here: indeed, while the resulting two random variables U, V *could* be made to have many matching moments, and as such be hard to distinguish, doing so will put a lot of probability mass on the roots of the Chebyshev polynomial T_d both for U and V; and these roots are not multiples of $1/m$. This would have the undesirable effect of making *both* our distributions far from being m-grained. Trying to fix this the obvious requires to ensure very little mass is

put by U (and so V) on the roots of T_d, which in turns limits the number of moments that can be matched. (Note that fixing this by choosing not to use the Chebyshev polynomial T_d at all but instead choosing $P(x)$ of the form $P(x) = x\left(x - \frac{1}{2m}\right) \cdot \left(x - \frac{1}{m}\right) \cdots \left(x - \frac{L}{m}\right)$, for some large constant integer L, does get part of the way there and provides a non-trivial result, leading to our (weaker) $\Omega(m^c)$ lower bound for every $c < 1$).

To remedy this, we take a different route, using an idea we believe to be of independent interest and applicable to other lower bounds: instead of using Chebyshev polynomials directly, we will be using a *modified* version of Chebyshev polynomials, $\tilde{T}_d$, defined from T_d by first rounding its roots to multiples of $1/m$. By carefully analyzing the resulting polynomial, we can show that it behaves, for our lower bound construction purposes, similarly to the Chebyshev polynomial, and so we can use it to define both U and V. This allows us to match more moments, leading to our final $\Omega(m/\log m)$ lower bound.

Upper bound for parameterized uniformity testing

In contrast, the idea underlying our upper bound for testing Π_m in the DoHO model is relatively straightforward: namely, by making n queries per sample, we can simulate any testing algorithm in the *standard* sampling model, as we now have observed the full samples. This, along with the relation between TV distance and EMD distance, allows us to lift any s-sample tester for Π_m in the standard sampling model to an s-sample, $s \cdot n$-query tester for Π_m in the DoHO model.

The only issue with this plan is that *there is no known (nontrivial) tester for* Π_m *in the standard sampling model.* There is, however, a known testing algorithm for the related property of generalized uniformity, Π_U. Our key contribution here is then to use this known tester A to derive a tester B for Π_m in the standard sampling model. Notably, this is not as immediate as it seems, and our algorithm needs to use A in a whitebox way, and combine this with an additional test to estimate the ℓ_2 norm of the unknown distribution D. The subtlety here (and the reason for which we cannot use *any* A in a blackbox fashion, but need to use a specific algorithm due to [5]) is that being close to *some* uniform distribution over *some* subset does not immediately allow to conclude about being close to *some* uniform distribution over some m*-size* subset – even if we are guaranteed the ℓ_2 norm of D is close to $1/m$.

1.2 Related work

The field of distribution testing has its roots in theoretical computer science with the work of Goldreich and Ron [21], who designed a uniformity testing algorithm as a tool to check whether a graph is an expander; and formally defined and introduced in [7]. [6] studied the problem of identity testing, which generalizes the problem of uniformity testing. Over the last two decades, this field has seen significant growth, and a host of new tools and techniques have emerged, see [25, 1, 18, 16, 14, 15, 11] to name a few. See the surveys [8, 4, 9] and the book of [19, Chapter 11] for more details.

The study of grained distributions was done implicitly in the work of [26], and later [18] studied it in more detail.

As mentioned earlier, the Huge Object model was introduced by [23]. Since then, there have been several works focusing on this model. [13] defined the notion of *index-invariant properties*, a set of properties that are invariant under the permutation of the indices (these properties are different than label-invariant properties). They showed that index-invariant properties whose VC dimension of the support set is bounded can be learned using a constant

number of queries, depending only on the VC dimension. They also gave tight separation between adaptive and non-adaptive testers for index and non-index-invariant properties. Later, [2] studied various different notions of adaptivity, and showed several separation results. Very recently, [3] studied the problem of support size testing in this model.

1.3 Preliminaries

For a positive integer n, let $[n]$ denote the set $\{1, \ldots, n\}$. We will use the standard asymptotic notation $\mathcal{O}(\cdot), \Omega(\cdot), \Theta(\cdot)$, and, in some cases, the (somewhat less) standard notation $\widetilde{O}(\cdot)$, which omits poly-logarithmic dependencies in the parameters for readability.

We will use several notions of distance between two distributions.

▶ **Definition 1.5.** *Let D_1 and D_2 be two probability distributions over a domain Ω. The ℓ_1* distance *between D_1 and D_2 is defined as*

$$\|D_1 - D_2\|_1 = \sum_{x \in \Omega} |D_1(x) - D_2(x)|.$$

The total variation distance *between D_1 and D_2 is defined as:*

$$\mathsf{d}_{\mathsf{TV}}(D_1, D_2) = \frac{1}{2} \cdot \|D_1 - D_2\|_1 = \sup_{S \subseteq \Omega} (D_1(S) - D_2(S)).$$

▶ **Definition 1.6** (EMD with respect to Hamming distance)**.** *Let D_1, D_2 be two distributions defined over the n-dimensional Hamming cube $\{0,1\}^n$ and d_H be the (relative) Hamming distance over $\{0,1\}^n$. Then the* Earth Mover distance *(EMD) between D_1 and D_2 is defined as follows:*

$$\mathsf{d}_{\mathsf{EM}}(D_1, D_2) := \inf_{T \in \mathcal{T}(D_1, D_2)} \mathbb{E}_{(x,y) \sim T} \left[d_H(x, y) \right]$$

where $\mathcal{T}(D_1, D_2)$ denotes the set of all couplings between μ and τ.

Since we are working with the (relative) Hamming distance, $d_H(x, y) \leq 1$. Then directly from the definitions of $\mathsf{d}_{\mathsf{EM}}(D_1, D_2)$ and $d_H(x, y)$, we get the following simple yet useful observation connecting total variation and EMD distances between two distributions.

▶ **Observation 1.7** (Relation between EMD and TV distances)**.** *Let D_1 and D_2 be two distributions over the n-dimensional Hamming cube $\{0,1\}^n$. Then,*

$$\mathsf{d}_{\mathsf{EM}}(D_1, D_2) \leq \mathsf{d}_{\mathsf{TV}}(D_1, D_2).$$

▶ **Definition 1.8** (Huge Object Model)**.** *Consider a discrete distribution D supported over the n-dimensional Hamming cube $\{0,1\}^n$. D is accessed by obtaining iid samples, where each sample obtained is an n-bit Boolean string. In addition to the sampling oracle, there is also a query oracle associated with D, where the tester can query any index $i \in [n]$ for any samples (say j-th sample s_j) obtained, and the query oracle will return the i-th bit of s_j. The goal is then to minimize both the total sample and query complexities required to decide a property.*

Note that this Huge Object model is particularly suited for studying high-dimensional distributions, where the dimension of the underlying domain is so large that even reading a single sample in its entirety might not be feasible.

Finally, we state here some technical result which will be used in our lower bounds proofs:

▶ **Lemma 1.9** ([31, Lemma 4])**.** *Let W_1, W_2 be discrete random variables taking values in $[0, \Lambda]$. If $\mathbb{E}[W_1^t] = \mathbb{E}[W_2^t]$ for $1 \leq t \leq T$, then*

$$\mathsf{d}_{\mathsf{TV}}\Big(\mathbb{E}_{W_1}[\mathsf{Poi}(W_1)], \mathbb{E}_{W_2}[\mathsf{Poi}(W_2)]\Big) \leq \left(\frac{e\Lambda}{2T}\right)^T.$$

▶ **Fact 1.10** ([10, Fact 7], [24]). *Let p be a degree-d polynomial with distinct roots $r_1, \ldots, r_d$. Then, for every $0 \le k \le d-2$, we have that $\sum_{i=1}^{d} \frac{r_i^k}{p'(r_i)} = 0$.*

We will be using the following two well-known trigonometric inequalities.

▶ **Fact 1.11.**

(i) *For $x \in [0, \pi]$, the following holds:*

$$\sin x \ge \frac{2}{\pi} \min(x, \pi - x) \tag{1}$$

(ii) *For $x \in [-\pi/2, \pi/2]$, the following holds:*

$$|\frac{4}{\pi} \cdot x| \ge |\sin x| \ge |\frac{2}{\pi} \cdot x|, \tag{2}$$

Organization

The rest of the paper is organized as follows. In Section 2, we present a new proof of the lower bound of $\Omega(m^c)$ samples. Then in Section 3, we present the $\Omega(m/\log m)$ lower bound for testing m-grainedness. In Section 4, we present our result of uniformity testing in the Huge Object Model. We conclude in Section 5 with some open questions. Due to the shortage of space, the formal proofs of some claims are deferred to the full version of the paper [12].

2 An $\Omega(m^c)$ lower bound for testing m-grainedness

In this section, we prove Theorem 1.2, thereby presenting a new proof of the lower bound of [22].

▶ **Theorem 1.2.** *Let n be a sufficiently large integer, $m = O(n)$, and let $\varepsilon \in (0,1)$ be a sufficiently small constant. Then, for any fixed constant $c \in (0,1)$, ε-testing m-grainedness of distributions over $[n]$ requires $\Omega(m^c)$ samples.*

Note that it suffices to consider the case $m = \Theta(n)$, as if $m \ll n$ one can then embed the hard instances in a larger domain, lifting the lower bound. As a result, we hereafter assume without loss of generality that $m = \Theta(n)$. We start by stating a relatively standard theorem which will be crucial in the proof of our sample complexity lower bound, and whose proof is deferred to the full version [12].

▶ **Theorem 2.1.** *Fix positive integers m, n, s, where $n > 4.3 \times 10^8$ and $m \le C_0(n-1)$ for some absolute constant $C_0 \in \mathbb{R}$. Suppose there exist random variables U and V, where U is supported on $\{0, \frac{1}{m}, \ldots, \frac{m}{m}\}$; $\Pr\left[V \in \{\frac{1}{2m}, \frac{3}{2m}, \ldots, \frac{2m-1}{2m}\}\right] \ge \frac{2}{3}$; and they satisfy the following three conditions,*

$$\max(U, V) \le \frac{20 \log^2(n-1)}{(n-1)}, \tag{3}$$

$$\mathbb{E}[U] = \mathbb{E}[V] \le \frac{1}{2(n-1)}. \tag{4}$$

$$\mathsf{d}_{\mathsf{TV}}\left(\mathbb{E}[\mathrm{Poi}(sU)], \mathbb{E}[\mathrm{Poi}(sV)]\right) \le \frac{1}{20(n-1)}, \tag{5}$$

Then any tester taking less than $s/2$ number of samples from unknown distribution p cannot distinguish between the cases that p is m-grained and p is at least $\frac{1}{8C_0}$-far from any m-grained distributions in TV distance with probability $3/5$.

In order to prove Theorem 1.2, we will be using the following lemma.

▶ **Lemma 2.2.** *There exist constants L, K, such that for any n and $m = L(n-1)$, there exist two discrete random variables U and V such that the following Equations* (6), (7), (8) *and* (9) *hold.*

$$\mathsf{Support}(U) \subseteq \frac{\mathbb{Z}}{m} \text{ and } \mathsf{Support}(V) \subseteq \frac{2\mathbb{Z}+1}{2m} \tag{6}$$

$$U, V \leq \frac{L}{m} \tag{7}$$

$$\mathbb{E}[U] = \mathbb{E}[V] \leq \frac{1}{2(n-1)}. \tag{8}$$

$$\mathbb{E}[U^t] = \mathbb{E}[V^t] \quad \text{for} \quad t \in [K]. \tag{9}$$

In order to prove the above lemma, we will use the following polynomial to construct our U and V (fix any positive integer L):

$$P(x) = x\left(x - \frac{1}{2m}\right)\left(x - \frac{2}{2m}\right)\left(x - \frac{3}{2m}\right)\cdots\left(x - \frac{2L}{2m}\right) \tag{10}$$

In the context of Lemma 2.2, the random variable U will be supported on the roots $0, \frac{1}{m}, \frac{2}{m}, \ldots, \frac{L}{m}$. Similarly, the random variable V will be supported on the roots $\frac{1}{2m}, \frac{3}{2m}, \ldots, \ldots, \frac{2L-1}{2m}$.

Proof of Lemma 2.2. We will first describe the construction of random variables U and V, associated with the polynomial P. Namely, let $\Pr[U = r] \propto \frac{1}{|P'(r)|}$ for r in the support of U (every derivative is positive when r is in the support of U, i.e., $r \in \{0, \frac{1}{m}, \frac{2}{m}, \ldots, \frac{L}{m}\}$). Similarly, we have $\Pr[V = r] \propto \frac{1}{|P'(r)|}$ for r in the support of V (likewise, all derivative will be negative). The normalization term is simply

$$Z_P := \sum_{i=0}^{L} \frac{1}{|P'(\frac{i}{m})|} = \sum_{i=1}^{L} \frac{1}{|P'(\frac{2i-1}{2m})|}.$$

And we can show that

$$Z_P = \sum_{k=0}^{L} \frac{\binom{2L}{2k}(2m)^{2L}}{(2L)!} = \frac{(2m)^{2L}}{(2L)!}\sum_{k=0}^{L}\binom{2L}{2k} = \frac{(2m)^{2L}}{(2L)!}\cdot 2^{2L-1}.$$

Applying Fact 1.10 with $k = 0$, we can notice that the two summations are equal (U has all positive roots and V all negative).

Proof of Equation (7). The largest value of U and V is the largest root of the polynomial P: $\max\{U, V\} \leq L/m$.

Proof of Equation (8). Let us compute the derivative of the polynomial P in Equation (10). For any $k \in [2L] \cup \{0\}$, we have the following:

$$\begin{aligned} P'\left(\frac{k}{2m}\right) &= \prod_{\ell=0}^{k-1}\frac{k-\ell}{2m}\prod_{\ell=k+1}^{2L}\frac{\ell-k}{2m}(-1)^{2\ell-k} \\ &= \frac{(-1)^k}{(2m)^{2L}}k!(2L-k)! \\ &= (-1)^k\frac{(2L)!}{\binom{2L}{k}(2m)^{2L}} \end{aligned}$$

Now let us compute $\mathbb{E}[U]$ as follows:

$$\begin{aligned}\mathbb{E}[U] &= \Pr[U=0]\cdot 0 + \sum_{\ell:\ \ell \text{ even},\ \ell\in[2m]} \Pr\left[U=\frac{\ell}{m}\right]\cdot\frac{\ell}{m}\\ &= \frac{1}{Z_P}\sum_{k=0}^{L}\frac{2k}{2m}\frac{\binom{2L}{2k}(2m)^{2L}}{(2L)!}\\ &= \frac{1}{m}\frac{\sum_{k=0}^{L}k\binom{2L}{2k}}{2^{2L-1}}\\ &= \frac{L}{2m}\end{aligned}$$

We finally upper bound the expectation:

$$\frac{1}{2(n-1)} \geq \frac{L}{2m} = \mathbb{E}[U] \Rightarrow m \geq L(n-1). \tag{11}$$

Proof of Equation (9). Their first $K = 2L-1$ moments are matched by recalling Fact 1.10 (applied for $k = t$) and how U and V are constructed. ◀

Proof of Theorem 1.2. Using Lemma 1.9 on random variables constructed in Lemma 2.2, by setting $m = 2L(n-1) = \Theta(n)$, we know $\Lambda = s\cdot\frac{L}{m}$, we get that

$$\mathsf{d}_{\mathsf{TV}}(\mathbb{E}[\mathrm{Poi}(sU),\mathrm{Poi}(sV)]) \leq \left(\frac{es\cdot\frac{L}{m}}{2K}\right)^{K} \leq \frac{1}{2(n-1)},$$

which gives,

$$s \leq 2(n-1)\left(\frac{2K}{e}\right)\cdot\left(\frac{1}{2(n-1)}\right)^{1/K}.$$

Further simplification yields,

$$s \leq \left(\frac{4L-2}{e}\right)(2(n-1))^{1-\frac{1}{2L-1}}.$$

Applying Theorem 2.1, gives the lower bound $s = \Omega\left(n^{1-\frac{1}{2L-1}}\right)$. For any constant $c < 1$, one can choose L large enough such that $c \leq 1-\frac{1}{2L-1}$, and thus concludes our proof. ◀

3 $\Omega(\frac{m}{\log m})$ lower bound for m-grainedness (Proof of Theorem 1.1)

In this section, we will be proving the following theorem:

▶ **Theorem 1.1.** *Let n be a sufficiently large integer, $m = O(n)$, and let $\varepsilon \in (0,1)$ be a sufficiently small constant. Then, ε-testing m-grainedness of distributions over $[n]$ requires $\Omega(\frac{m}{\log m})$ samples.*

Similar to the proof of Theorem 1.2, we will make use of Theorem 2.1 to prove our $\Omega(\frac{m}{\log m})$ lower bound. Namely, we will construct two random variables U and V such that they can be used to construct approximate probability vector for m-grained and mostly $2m$-grained and their first $\log(m)$ moments match. In particular, we will prove the following lemma:

▶ **Lemma 3.1.** *For $m = 70(n-1)$ and $n > 20$, there exist discrete random variables U and V such that the following hold:*

$$\Pr[V = \frac{1}{2m}] \geq 2/3. \tag{12}$$

$$\mathsf{Support}(U) \subseteq \frac{\mathbb{Z}}{m} \tag{13}$$

$$0 \leq U, V \leq \frac{20\log^2(n-1)}{n-1}. \tag{14}$$

$$\mathbb{E}[U] = \mathbb{E}[V] \leq \frac{1}{20(n-1)}. \tag{15}$$

$$\mathbb{E}[U^t] = \mathbb{E}[V^t] \text{ for } \ t \in [3 \cdot \log(n-1)] \tag{16}$$

3.1 Construction of U and V

The main focus of this subsection is to establish Lemma 3.1.

Fact 1.10 allows us to define two random variables U and V with matching moments up to the degree of some polynomial – similar to the analysis in Section 2, by simply defining U and V through its roots and partitioning them by the signs of the derivatives at those roots. However, the challenge here is, we need to match even higher degree (in Section 2, we can match them up to a constant degree C_1, here we need to match them up to $\Theta(\log m) = \Theta(\log n)$ degree), all while keeping every other conditions unchanged: U is m-grained and V on the most part is $(2m)$-grained; maximum value of the support size does not blow up; and more importantly, the expectation could be bounded by $O(1/n)$. Notably, if we try to match higher degree: say instead of constant, we make $L = O(\log n)$ in Equation (11), with $m = \Theta(n)$, using the construction in Section 2. It will fail miserably – the expectation $\mathbb{E}[U]$ will exceed $O(1/n)$, and so when we take n copies of U or V, it would not be a distribution in an approximate sense.

To work around this and match degree (by implication, moments) as high as possible, we take the (shifted) Chebyshev polynomial of degree d and we will later use properties of this polynomial to construct our prior variables U and V. Recall that the shifted Chebyshev polynomial of the first kind is defined as follows:

$$p_{T_d}(x) = T_d(1 - \Delta x) = 2^{d-1}\Delta^d \cdot \prod_{i=1}^{d}(x - t_i), \tag{17}$$

where $T_d(\theta) = \cos(d \cdot \arccos\theta)$, for $\theta \in [-1, 1]$; t_i denotes the roots of the degree-d (shifted) Chebyshev polynomial for every $i \in [d]$; $\Delta \in \mathbb{R}$, a parameter to be chosen in the analysis later.

However, as mentioned in the introduction, instead of using the shifted Chebyshev polynomial directly, we will be using a variant of it for our proof. For this purpose, we will define the polynomial $\tilde{p}_{T_d}$, a slightly modified Chebyshev polynomial of degree d, by "rounding up to the nearest multiple of $1/m$" its d roots:

$$\tilde{p}_{T_d}(x) = \tilde{T}_d(1 - \Delta \cdot x) = 2^{d-1}\Delta^d \cdot \prod_{i=1}^{d}(x - \tilde{t}_i) \tag{18}$$

where $\frac{1}{m} + t_i \geq \tilde{t}_i = \frac{1}{m}\lceil mt_i \rceil \geq t_i$.

Putting these ideas together, we will focus on two polynomials based on the Chebyshev polynomial and modified Chebyshev polynomial respectively (obtained by appending Chebyshev polynomial to $x(x-\frac{1}{2m})(x-\frac{1}{m})$):

$$p(x) = x\left(x-\frac{1}{2m}\right)\left(x-\frac{1}{m}\right)\cdot p_{T_d}(x) \tag{19}$$

$$\tilde{p}(x) = x\left(x-\frac{1}{2m}\right)\left(x-\frac{1}{m}\right)\cdot \tilde{p}_{T_d}(x) \tag{20}$$

Through polynomial $\tilde{p}$ in Equation (20), we can construct two random variables U and V identified by their probability mass function. Namely, we define U as follows:

$$\Pr[U=r] \propto \frac{1}{|\tilde{p}'(r)|} \quad \text{if } \tilde{p}'(r)>0 \text{ and } r \text{ is a root of } \tilde{p} \tag{21}$$

Similarly, we define the random variable V as follows:

$$\Pr[V=r] \propto \frac{1}{|\tilde{p}'(r)|} \quad \text{if } \tilde{p}'(r)<0 \text{ and } r \text{ is a root of } \tilde{p} \tag{22}$$

For both U and V, we assign probability 0 to the roots not specified. Thus, by setting $k=0$ in Fact 1.10 and noting that the negative and positive terms sum to 0, the normalization factor are equal for both U and V, and can be expressed as follows:

▶ **Observation 3.2.**

$$Z_{\tilde{p}} = Z(\Delta,m,d) = \frac{1}{\tilde{p}'(0)} + \frac{1}{\tilde{p}'(\frac{1}{m})} + \sum_{j=1}^{d}\frac{\mathbb{1}_{\{\tilde{p}'(\tilde{t}_j)>0\}}}{\tilde{p}'(\tilde{t}_j)} = \frac{1}{|\tilde{p}'(\frac{1}{2m})|} + \sum_{j=1}^{d}\frac{\mathbb{1}_{\{\tilde{p}'(\tilde{t}_j)<0\}}}{|\tilde{p}'(\tilde{t}_j)|}. \tag{23}$$

The reasons we are taking the shifted Chebyshev polynomial and "round the roots up" are two-fold: first, this kind of construction (before rounding) was used to prove lower bounds with $\Omega(n/\log n)$ complexity before, by constructing a polynomial of degree $\Theta(\log n)$ with certain constraints [10]. Second, we need to make sure that the distributions derived from U are m-grained with very high probability. Yet, using the shifted Chebyshev directly, some of the roots of U (the positive roots) resulting from T_d will not be m-grained with non-trivial probability. Intuitively, we want to leverage the known properties of the shifted Chebyshev p_{T_d} and argue that rounding up and creating $\tilde{p}_{T_d}$ will only "mildly" change those properties.

Since we are deciding the support of U, V based on the evaluation of derivative at each root of $\tilde{p}_{T_d}$ and argue that it is somewhat close to p_{T_d}, we need to first make sure that the signs of derivative at the main three roots remain the same as p_{T_d}'s (the rest of the roots' derivative's signs, the ones induced by $\tilde{p}_{T_d}$, will not affect the argument). Note that the sign of the polynomial $\tilde{p}_{T_d}$ when evaluated at $0, 1/2m, 1/m$ is not changed (same as p_{T_d}), as $(0-t_j)$ and $(0-\tilde{t}_j)$ have the same sign; we also have $1/2m, 1/m \le t_j \le \tilde{t}_j$, via a constraint on setting Δ and therefore we can conclude that

$$p_{T_d}(0)\cdot\tilde{p}_{T_d}(0)>0; \quad p_{T_d}(1/2m)\cdot\tilde{p}_{T_d}(1/2m)>0; \quad p_{T_d}(1/m)\cdot\tilde{p}_{T_d}(1/m)>0.$$

Next, we compute their corresponding derivatives of the two polynomials evaluated at the roots t_ℓ and $\tilde{t}_\ell$.

$$p'(t_\ell) = t_\ell\left(t_\ell-\frac{1}{2m}\right)\left(t_\ell-\frac{1}{m}\right)2^{d-1}\Delta^d\prod_{j\neq\ell}(t_j-t_\ell) \tag{24}$$

Table 1 Properties of the polynomial p and its distinct $d+3$ roots. $|p'(r_1)|, |p'(r_2)|, |p'(r_3)|$'s upper and lower bounds are proven in Claim A.7 of the full version [12].

roots	value of $p'(r_\ell)$	bound on $\lvert p'(r_\ell)\rvert$
$r_1 = 0$	$\frac{1}{2m^2} \cdot T_d(1)$	$\Theta\left(\frac{1}{m^2}\right)$
$r_2 = \frac{1}{2m}$	$-\frac{1}{4m^2} \cdot T_d\left(1 - \frac{\Delta}{2m}\right)$	$\Theta\left(\frac{1}{m^2}\right)$
$r_3 = \frac{1}{m}$	$\frac{1}{2m^2} \cdot T_d\left(1 - \frac{\Delta}{m}\right)$	$\Theta\left(\frac{1}{m^2}\right)$
$r_{\ell+3} = t_\ell = \frac{2}{\Delta}\sin^2\left(\frac{\pi}{2}\left(\frac{2\ell-1}{2d}\right)\right)$	$p'(t_\ell) = t_\ell \cdot \left(t_\ell - \frac{1}{2m}\right) \cdot \left(t_\ell - \frac{1}{m}\right) \cdot T'_d(t_\ell)$	$\Theta\left(\frac{\ell^5}{\Delta^2 d^4}\right)$

$$\tilde{p}'(\tilde{t}_\ell) = \tilde{t}_\ell \left(\tilde{t}_\ell - \frac{1}{2m}\right)\left(\tilde{t}_\ell - \frac{1}{m}\right) 2^{d-1}\Delta^d \prod_{j\neq\ell}(\tilde{t}_j - \tilde{t}_\ell) \tag{25}$$

Here, note that by construction, all the roots of $\tilde{p}_{T_d}$ will be m-grained (see Equation (18)). Moreover, 0 and $\frac{1}{m}$ are m-grained roots of $\widetilde{p}(x)$ (see Equation (20)). On the other hand, the root $\frac{1}{2m}$ of the polynomial $\widetilde{p}(x)$ is not m-grained by definition.

In the context of Lemma 3.1, the random variable U will be supported on the roots 0, $\frac{1}{m}$, and a subset of the roots $\widetilde{t}_\ell$ of the modified Chebyshev polynomial $\widetilde{p}_{T_d}$ from Equation (18), for $\ell \in [d]$. On the other hand, the random variable V will be supported over $\frac{1}{2m}$ and a disjoint (from support of U constructed by modified Chebeshev roots of $\tilde{p}_{T_d}$) subset of the modified Chebyshev polynomial roots. U and V's support form a partition for all roots of $\tilde{p}_{T_d}$.

To prove Lemma 3.1, we need a few claims, that we state below and whose proof can be found in the full version [12].

▷ Claim 3.3 (Relation between roots at $\tilde{p}'$ and p').

$$\frac{|\tilde{p}'(\tilde{t}_\ell)|}{|p'(t_\ell)|} \geq \exp\left(-\frac{12\Delta}{m/d^2}\right) \text{ for every } \ell \in [d]. \tag{26}$$

$$1 \leq \frac{|\tilde{p}'(0)|}{|p'(0)|}, \quad 1 \leq \frac{|\tilde{p}'(1/m)|}{|p'(1/m)|} \leq 6, \quad 1 \leq \frac{|\widetilde{p}'(1/2m)|}{|p'(1/2m)|} \leq 6. \tag{27}$$

The proofs of Equation (26) and Equation (27) can be found in Appendix A.1 of the full version (Claims A.2 to A.5). From now onwards, we will assume that Claim 3.3 holds.

For the lower bound, we will be choosing $\Delta = \frac{m}{20d^2}$ and $d = \sqrt{10}\log(n-1)$.

▶ **Observation 3.4.** *Let t_j and $\tilde{t}_j$ be the roots of the Cheybyshev and modified Chebyshev polynomials, for any $j \in [d]$. When $\Delta \leq \frac{m}{2d^2}$, we have that*

$$\tilde{t}_j \geq t_j \geq \frac{1}{m}.$$

Proof. Using Equation (2), we can lower bound t_j for any j,

$$t_j = \frac{1}{\Delta}\left(1 - \cos\left(\frac{2j-1}{2d}\pi\right)\right) = \frac{2}{\Delta}\sin^2\left(\frac{\pi}{2}\left(\frac{2j-1}{2d}\right)\right) \geq \frac{2}{\Delta}\left(\frac{2j-1}{2d}\right)^2 \geq \frac{1}{m}, \tag{28}$$

where the last inequality follows from $2j-1 \geq 1$ (which holds for all $j \geq 1$), and our assumption on Δ. Moreover, we also have:

$$\tilde{t}_j = \frac{1}{m}\lceil mt_j \rceil \geq t_j \geq \frac{1}{m}$$

where the last inequality follows from Equation (28). This completes the proof. ◀

We need the following two claims (Claim 3.5 and Claim 3.6) whose proofs are deferred to Appendix A.2 of the full version [12], to show that $\Pr[V = \frac{1}{2m}] \geq \Omega(1)$ in Claim 3.7 as required by Equation (12).

▷ Claim 3.5. Suppose $1 \leq \frac{|\widetilde{p'}(\frac{1}{2m})|}{|p'(\frac{1}{2m})|} \leq 6$ holds and $\Delta \leq \frac{2m}{9d^2}$. Then we have,

$$\Pr\left[V = \frac{1}{2m}\right] \propto \frac{1}{|\tilde{p}'(\frac{1}{2m})|} \geq \frac{2}{3}m^2.$$

▷ Claim 3.6. Suppose that $\Delta \leq \frac{2m}{9d^2}$. For any $j \in [d]$, let $\tilde{t}_j$ denote the j-th root of the modified Chebyshev polynomial $\widetilde{p}$. Then we have:

$$\Pr\left[V \neq \frac{1}{2m}\right] \propto \sum_{j=1}^{d} \frac{\mathbb{1}_{\{\tilde{p}'(\tilde{t}_j)<0\}}}{|\tilde{p}'(\tilde{t}_j)|} \leq \exp\left(\frac{12\Delta}{m/d^2}\right) \cdot \frac{66}{\pi} \cdot \Delta^2 d^4.$$

▷ Claim 3.7. If $\Delta \leq \frac{m}{20d^2}$, then

$$\Pr\left[V = \frac{1}{2m}\right] = \frac{1}{|\tilde{p}'(\frac{1}{2m})| \cdot Z} \geq \frac{5}{6},$$

where Z is the normalizing constant defined in Equation (23).

Proof. Following the definition of the random variable V (Equation (22)) and the normalization term $Z_{\tilde{p}}$ (Equation (23)), we can say that:

$$\Pr\left[V = \frac{1}{2m}\right] = \frac{1}{|\tilde{p}'(\frac{1}{2m})| \cdot Z} = \frac{1}{1 + \left(\sum_{j=1}^{d} \frac{\mathbb{1}_{\{\tilde{p}'(\tilde{t}_j)<0\}}}{|\tilde{p}'(\tilde{t}_j)|}\right) \cdot |\tilde{p}'(\frac{1}{2m})|}.$$

Combining Claim 3.5 and Claim 3.6, we see that

$$\begin{aligned}
\left(\sum_{j=1}^{d} \frac{\mathbb{1}_{\{\tilde{p}'(\tilde{t}_j)<0\}}}{|\tilde{p}'(\tilde{t}_j)|}\right) \cdot |\tilde{p}'(\frac{1}{2m})| &\leq \frac{3\exp\left(\frac{12\Delta}{m/d^2}\right)}{2m^2} \cdot \frac{66}{\pi} \cdot \Delta^2 d^4 \\
&= \frac{99}{\pi}\exp\left(\frac{12\Delta}{m/d^2}\right) \cdot \frac{\Delta^2 d^4}{m^2} \\
&\leq \frac{99}{\pi}\exp\left(\frac{12\frac{m}{20d^2}}{m/d^2}\right) \cdot \frac{\left(\frac{m}{20d^2}\right)^2 d^4}{m^2} \quad [\because \Delta \leq \frac{m}{20d^2}] \\
&\leq \frac{99}{\pi}\exp(\frac{12}{20}) \cdot \frac{1}{400} \leq \frac{1}{5}.
\end{aligned}$$

Thus, we can say that

$$\Pr\left[V = \frac{1}{2m}\right] = \frac{1}{1 + \left(\sum_{j=1}^{d} \frac{\mathbb{1}_{\{\tilde{p}'(\tilde{t}_j)<0\}}}{|\tilde{p}'(\tilde{t}_j)|}\right) \cdot |\tilde{p}'(\frac{1}{2m})|} \geq \frac{5}{6}.$$

This completes the proof. ◁

Now we are ready to prove Lemma 3.1.

Proof of Lemma 3.1. We now prove that the random variables U and V defined in Section 3.1 satisfies Equation (12)-Equation (16). We begin by setting $\Delta = \frac{m}{20d^2}$.

Proof of Equation (12). Using Claim 3.7, we know that

$$\Pr\left[V = \frac{1}{2m}\right] \geq 5/6.$$

Proof of Equation (14). The largest value of U and V are the largest root of the modified Chebyshev polynomial. Now let us first upper bound the largest value of the roots of the Chebyshev polynomial. The largest value of the roots of Chebyshev polynomial is:

$$\max_{\ell\in[d]} t_\ell = \max_{\ell\in[d]} \frac{1}{\Delta}\left(1 - \cos\left(\frac{2\ell-1}{2d}\pi\right)\right) = \frac{2}{\Delta}\sin^2\left(\frac{\pi}{2}\left(\frac{2d-1}{2d}\right)\right) \leq \frac{2}{\Delta},$$

where the last inequality is obtained via Equation (2). So, the largest value of the roots of the modified Chebyshev polynomial is:

$$\max_{\ell\in[d]} \tilde{t}_\ell = \frac{1}{m}\lceil mt_j\rceil \leq \frac{2}{\Delta} + \frac{1}{m} = \frac{40d^2+1}{m},$$

The last equality follows from the fact that $\Delta = \frac{m}{20d^2}$. Plugging the values of $m = 70(n-1)$ and $d = \sqrt{10}\log(n-1)$, we have the result.

Proof of Equation (15). Recall that the random variable U is supported on the roots 0, $1/m$ and a subset of the roots of the modified Chebyshev polynomial (Equation (18)). Let us start by computing an upper bound on $\mathbb{E}[U]$.

$$\begin{aligned}
\mathbb{E}[U] &\leq \Pr[U=0]\cdot 0 + \Pr[U=\frac{1}{m}]\cdot\frac{1}{m} + \sum_{j=1}^{d}\Pr[U=\tilde{t}_j]\cdot\tilde{t}_j \\
&= \Pr[U=\frac{1}{m}]\cdot\frac{1}{m} + \frac{1}{Z_{\tilde{p}}}\left(\sum_{j=1}^{d}\tilde{t}_j\cdot\frac{1}{|\tilde{p}'(\tilde{t}_j)|}\right) \quad [Z_{\tilde{p}} \text{ is the normalization term, see Equation (23)}] \\
&\leq \frac{1}{m} + \frac{1}{Z_{\tilde{p}}}\left(\sum_{j=1}^{d}\left(t_j+\frac{1}{m}\right)\cdot\frac{\exp\left(\frac{12\Delta}{m/d^2}\right)}{|p'(t_j)|}\right) \qquad \left[\because \tilde{t}_j = \frac{1}{m}\lceil mt_j\rceil \text{ and Equation (26)}\right] \\
&\leq \frac{1}{m} + \frac{1}{Z_{\tilde{p}}}\left(\sum_{j=1}^{d}\left(1+\frac{1}{2}\right)t_j\cdot\frac{\exp\left(\frac{12\Delta}{m/d^2}\right)}{\frac{\Delta}{4}\cdot t_j^3\cdot\frac{d}{\sin\frac{2j-1}{2d}}}\right) \\
&\qquad [\because \text{using the proof of Claim 3.6 (Appendix A.2, full version [12])}] \\
&= \frac{1}{m} + \frac{1}{Z_{\tilde{p}}}\left(\sum_{j=1}^{d}\frac{6\cdot\exp\left(\frac{12\Delta}{m/d^2}\right)}{\Delta\cdot\left(\frac{4}{\Delta^2}\sin^4\left(\frac{\pi}{2}\cdot\frac{2j-1}{2d}\right)\right)\cdot\frac{d}{\sin\frac{2j-1}{2d}}}\right) \qquad [\because t_j = \tfrac{2}{\Delta}\sin^2\left(\tfrac{\pi}{2}\left(\tfrac{2j-1}{2d}\right)\right)] \\
&\leq \frac{1}{m} + \frac{1}{Z_{\tilde{p}}}\left(\sum_{j=1}^{d}\frac{6\cdot\exp\left(\frac{12\Delta}{m/d^2}\right)}{\left(\frac{4}{\Delta}\cdot\left(\frac{2j-1}{2d}\right)^4\right)\cdot\frac{d}{\frac{4}{\pi}\cdot\frac{(2j-1)}{2d}}}\right) \qquad [\because \text{Via Equation (2)}] \\
&= \frac{1}{m} + \frac{\exp\left(\frac{12\Delta}{m/d^2}\right)\cdot\Delta d^2}{Z_{\tilde{p}}}\cdot\left(\sum_{j=1}^{d}\frac{6}{\pi(j-1/2)^3}\right) \\
&\leq \frac{1}{m} + 17\cdot\frac{\exp\left(\frac{12\Delta}{m/d^2}\right)\cdot\Delta d^2}{Z_{\tilde{p}}} \qquad \left[\because \sum_{i=1}^{\infty}\frac{6}{\pi\cdot(i-1/2)^3} \leq 17\right]
\end{aligned}$$

$$= \frac{1}{m} + \frac{17\times 3}{2}\cdot\frac{\exp\left(\frac{12\Delta}{m/d^2}\right)\cdot\Delta d^2}{m^2} \qquad \left[\because Z_{\tilde{p}} \geq \frac{1}{|\tilde{p}'(\frac{1}{2m})|} \geq \frac{2}{3}m^2 \text{ via Equation (23) \& Claim 3.5}\right]$$

$$\leq \frac{7}{2m} = \frac{1}{20(n-1)} \qquad \left[\because \Delta \leq \frac{m}{20d^2} \;\&\; m = 70(n-1)\right]$$

Since Equation (16) holds, as we will prove below, we know that $\mathbb{E}[U] = \mathbb{E}[V]$. This implies that $\mathbb{E}[V] \leq \frac{1}{20(n-1)}$ holds as well.

Proof of Equation (16). We have $\mathbb{E}[U^t] = \mathbb{E}[V^t]$ for $1 \leq t \leq d+1$ from Fact 1.10 and Observation 3.2. ◀

3.2 Putting it together: proof of the $\Omega(m/\log m)$ lower bound

We are now ready to prove the lower bound of m-grainedness testing (Theorem 1.1): we will combine the results from Lemma 3.1, Lemma 1.9 and Theorem 2.1 to show our main $\Omega(m/\log m)$ lower bound in Theorem 1.1.

Proof of Theorem 1.1. Let us fix $n > 20, m = 70(n-1)$, and $s \geq 1$ be a parameter (we will later on set it to $s = \Theta(n/\log n)$). From Lemma 3.1, we know that there exist two discrete random variables U and V with the matching moments properties. Now we will use Lemma 1.9 for sU, sV. Following Lemma 3.1 and Lemma 1.9, we have $\Lambda \leq s \cdot \frac{20\log^2(n-1)}{n-1}$ and $T = 3\cdot\log(n-1)$

$$\mathsf{d}_{\mathsf{TV}}\left(\mathbb{E}[\mathrm{Poi}(sU), \mathbb{E}[\mathrm{Poi}(sV)]]\right) \leq \left(\frac{e\cdot s\cdot\frac{20\log^2(n-1)}{n-1}}{3\cdot\log(n-1)}\right)^{3\cdot\log(n-1)}.$$

And we want to satisfy Equation (5), as all others have been matched to use Theorem 2.1.

$$\left(\frac{e\cdot s\cdot\frac{20\log^2(n-1)}{n-1}}{3\cdot\log(n-1)}\right)^{3\cdot\log(n-1)} \leq \frac{1}{20(n-1)} \Leftrightarrow \frac{e\cdot s\cdot 20\log(n-1)}{3(n-1)} \leq \left(\frac{1}{20(n-1)}\right)^{\frac{1}{3\cdot\log(n-1)}}$$

For $n \geq 21$, we have

$$\frac{e\cdot s\cdot 20\log(n-1)}{3(n-1)} \leq \frac{1}{4} \leq \left(\frac{1}{20(n-1)}\right)^{\frac{1}{3\cdot\log(n-1)}},$$

and thus for $s = \frac{3(n-1)}{80e\log(n-1)}$, we can invoke, and get a sample complexity lower bound in Theorem 1.1, noting that $m = 70(n-1) = \Theta(n)$. Thus

$$\frac{s}{2} = \frac{3(n-1)}{160e\log(n-1)} = \Omega\left(\frac{m}{\log m}\right).$$

This concludes our proof. ◀

4 Uniformity Testing in the DoHo Model

In this section, we prove the following result on parameterized uniformity testing in the DoHO model, improving on the known $\tilde{O}(m)$-query upper bound in the regime $m \geq n^3$.

▶ **Theorem 4.1.** *For any m and constant $\varepsilon > 0$, the property Π_m of distributions over $\{0,1\}^n$ defined as*

$$\Pi_m = \{U_S : S \subseteq \{0,1\}^n, |S| = m\}$$

can be ε-tested in the DoHO model using $s = \mathcal{O}(m^{2/3}/\varepsilon^6)$ samples and $s\cdot n$ queries. Moreover, $\Omega(m^{2/3})$ samples are necessary.

Proof. The algorithm itself is quite simple, and follows from combining existing algorithms for *generalized uniformity* testing in the standard sampling model [5, 17] with an additional "check" of the ℓ_2 norm of the distribution: the tester is described in Algorithm 1. The query complexity follows trivially from the stated sample complexity, as n queries suffice to read the full sample.

Specifically, the algorithm is as follows:

Algorithm 1 Algorithm to test Π_m.

1: Set $\delta \leftarrow 1/6$, and $\alpha = \Theta(\varepsilon)$.
2: Run the adaptive ℓ_2 norm estimator of [5, Lemma 3.1] to obtain an $(1 \pm \varepsilon/10)$ estimate $\hat{\rho}$ of $\|D\|_2^2$ (with error probability δ) using $O(\sqrt{m}/\varepsilon^2)$ samples. If the algorithm does not terminate with this number of samples, or if the estimate is not in $(1 \pm \varepsilon/10)/m$, output reject.
3: Run the generalized uniformity tester of [5] on s (fully queried) samples from D with (TV) distance parameter α and error probability δ. If it rejects, output reject.
4: Output accept.

We now argue correctness. By a union bound, both subroutines behave as they should with overall probability 2/3: we hereafter assume their output is correct.

- *Completeness:* Assume $D \in \Pi_m$. Then $\|D\|_2^2 = \frac{1}{m}$, so $\hat{\rho} \in [\frac{1-\varepsilon/10}{m}, \frac{1+\varepsilon/10}{m}]$ and the first step does not reject. Similarly, $D \in \Pi_U = \bigcup_{k=1}^{\infty} \Pi_k$, so the generalized uniformity tester (which is by definition a tester for Π_U) accepts. Overall, Algorithm 1 returns accept.
- *Soundness:* By contrapositive, suppose Algorithm 1 returns accept. Since the second subroutine did not reject, it must be the case that D is $\frac{\varepsilon}{2}$-close (in total variation distance) to a distribution U_T, uniform on some subset T of a given size k. Moreover, the algorithm of [5] provides the extra guarantee that $(1 - O(\varepsilon))\|D\|_2^2 \leq \frac{1}{k} \leq (1 + O(\varepsilon))\|D\|_2^2$ (see [5, Lemma 3.4]).[1]

  Now, let $U_S \in \Pi_m$ be a closest distribution to D, over all subsets S of size m:

$$\mathsf{d}_{\mathsf{TV}}(D, U_S) \leq \mathsf{d}_{\mathsf{TV}}(D, U_T) + \mathsf{d}_{\mathsf{TV}}(U_T, U_S) \leq \frac{\varepsilon}{2} + 1 - \frac{\min(m,k)}{\max(m,k)}$$

  where the second inequality follows the minimum TV distance between two uniform distributions on supports k and m, which occurs when the supports of the distributions overlap as much as possible, and is equal to $1 - \frac{\min(m,k)}{\max(m,k)}$. In particular, we have $1 - \frac{\min(m,k)}{\max(m,k)} \leq \frac{\varepsilon}{2}$ if

$$\frac{1}{1 - \frac{\varepsilon}{2}} \cdot m \leq k \leq (1 - \tfrac{\varepsilon}{2}) \cdot m$$

  Now, to see why this holds, observe that by our first check, $\|D\|_2^2$ is within a $(1 \pm \frac{\varepsilon}{10})$ factor of $\frac{1}{m}$, and, by the additional guarantee of the [5] tester (adjusting the constant in the setting of α), within a $(1 \pm \frac{\varepsilon}{10})$ factor of $\frac{1}{k}$. This implies that m and k are within a $1 \pm \frac{\varepsilon}{4}$ factor of each other, and thus that $1 - \frac{\min(m,k)}{\max(m,k)} \leq \frac{\varepsilon}{2}$. Overall, this establishes that $\mathsf{d}_{\mathsf{TV}}(D, U_S) \leq \varepsilon$.

Finally, the claimed lower bound follows directly from the lower bound on generalized uniformity testing of [17], which holds even when the target size of the support is given. ◀

[1] This is why we chose to use this specific algorithm, instead of that of [17], which has better sample complexity (with respect to ε) but does not provide this extra guarantee.

5 Conclusion and open problems

In this work, we studied the problem of testing m-grainedness of distributions. We established a new lower bound of $\Omega(m/\log m)$ samples, improving on the previous lower bound of $\Omega(m^c)$ due to [22], thereby settling a conjecture by [22]. Along the way, we also obtained an alternative, simpler proof of the $\Omega(m^c)$ lower bound. By leveraging a reduction between the testing models due to [23], our result implies an optimal lower bound for uniformity testing in the DoHO (Distributions over Huge Objects) model, settling another conjecture of [23]. Finally, we provided a simple tester for uniformity testing in this DoHO model, with improved sample complexity for a large range of the parameters.

Our work leaves open several new avenues of research in this context; we list two of them below:

- **(i)** Our lower bound for m-grainedness testing establishes the optimal dependence on the parameter m, for constant proximity parameter $\varepsilon = \Omega(1)$. It would be interesting to fully characterize the complexity of the question, including the dependence on ε.
- **(ii)** From [23], it is known that testing uniformity over an m-element subset of $\{0,1\}^n$ requires $\Omega(m/\log m)$ queries, for $m \leq n$; and that $\tilde{O}(m)$ queries are sufficient for all m. Our own tester (Theorem 4.1) shows an upper bound of $O(m^{2/3}n)$ queries, better in the regime $m \geq n^3$. This leaves understanding the landscape of parameterized uniformity testing, especially in the intermediate regime $n \ll m \ll n^3$, as an open and intriguing question.

References

1 Jayadev Acharya, Constantinos Daskalakis, and Gautam Kamath. Optimal testing for properties of distributions. *Advances in Neural Information Processing Systems (NeurIPS)*, 28, 2015.

2 Tomer Adar and Eldar Fischer. Refining the adaptivity notion in the huge object model. In *International Conference on Randomization and Computation (RANDOM)*, pages 45:1–45:16, 2024. doi:10.4230/LIPICS.APPROX/RANDOM.2024.45.

3 Tomer Adar, Eldar Fischer, and Amit Levi. Support testing in the huge object model. In *International Conference on Randomization and Computation (RANDOM)*, pages 46:1–46:16, 2024. doi:10.4230/LIPICS.APPROX/RANDOM.2024.46.

4 Sivaraman Balakrishnan and Larry Wasserman. Hypothesis testing for high-dimensional multinomials: A selective review, 2018.

5 Tugkan Batu and Clément L Canonne. Generalized uniformity testing. In *Symposium on Foundations of Computer Science (FOCS)*, pages 880–889, 2017. doi:10.1109/FOCS.2017.86.

6 Tugkan Batu, Eldar Fischer, Lance Fortnow, Ravi Kumar, Ronitt Rubinfeld, and Patrick White. Testing random variables for independence and identity. In *Symposium on Foundations of Computer Science (FOCS)*, pages 442–451, 2001. doi:10.1109/SFCS.2001.959920.

7 Tugkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D Smith, and Patrick White. Testing that distributions are close. In *Symposium on Foundations of Computer Science (FOCS)*, pages 259–269, 2000. doi:10.1109/SFCS.2000.892113.

8 Clément L Canonne. A survey on distribution testing: Your data is big. but is it blue? *Theory of Computing*, pages 1–100, 2020.

9 Clément L Canonne. Topics and techniques in distribution testing: A biased but representative sample. *Foundations and Trends® in Communications and Information Theory*, pages 1032–1198, 2022. doi:10.1561/0100000114.

10 Clément L Canonne, Ilias Diakonikolas, Daniel Kane, and Sihan Liu. Nearly-tight bounds for testing histogram distributions. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:31599–31611, 2022.

11 Clément L Canonne, Ayush Jain, Gautam Kamath, and Jerry Li. The price of tolerance in distribution testing. In *Conference on Learning Theory (COLT)*, pages 573–624, 2022. URL: https://proceedings.mlr.press/v178/canonne22a.html.

12 Clément L. Canonne, Sayantan Sen, and Joy Qiping Yang. Settling the complexity of testing grainedness of distributions, and application to uniformity testing in the huge object model. *ECCC preprint*, 2024. URL: https://eccc.weizmann.ac.il/report/2024/196.

13 Sourav Chakraborty, Eldar Fischer, Arijit Ghosh, Gopinath Mishra, and Sayantan Sen. Testing of index-invariant properties in the huge object model. In *Conference on Learning Theory (COLT)*, pages 3065–3136, 2023. URL: https://proceedings.mlr.press/v195/chakraborty23a.html.

14 Ilias Diakonikolas, Themis Gouleakis, John Peebles, and Eric Price. Sample-optimal identity testing with high probability. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, 2018.

15 Ilias Diakonikolas, Themis Gouleakis, John Peebles, and Eric Price. Collision-based testers are optimal for uniformity and closeness. *Chic. J. Theor. Comput. Sci*, 25:1–21, 2019. URL: http://cjtcs.cs.uchicago.edu/articles/2019/1/contents.html.

16 Ilias Diakonikolas and Daniel M Kane. A new approach for testing properties of discrete distributions. In *Symposium on Foundations of Computer Science (FOCS)*, pages 685–694, 2016. doi:10.1109/FOCS.2016.78.

17 Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Sharp bounds for generalized uniformity testing. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

18 Oded Goldreich. The uniform distribution is complete with respect to testing identity to a fixed distribution. In *Electronic Colloquium on Computational Complexity (ECCC)*, page 1, 2016.

19 Oded Goldreich. *Introduction to property testing*. Cambridge University Press, 2017. doi:10.1017/9781108135252.

20 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, pages 653–750, 1998. doi:10.1145/285055.285060.

21 Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. *Electron. Colloquium Comput. Complex.*, TR00-020, 2000. URL: https://eccc.weizmann.ac.il/eccc-reports/2000/TR00-020/index.html.

22 Oded Goldreich and Dana Ron. A lower bound on the complexity of testing grained distributions. *Computational Complexity (CC)*, page 11, 2023. doi:10.1007/S00037-023-00245-W.

23 Oded Goldreich and Dana Ron. Testing distributions of huge objects. In *TheoretiCS*, page 78, 2023.

24 metamorphy. Proving a formula for n-th degree polynomial with n distinct real roots, 2021. URL: https://math.stackexchange.com/questions/4074098/proving-a-formula-for-sum-j-1n-fracx-jkfx-j-for-f-an-n-th-degr.

25 Liam Paninski. A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Transactions on Information Theory*, pages 4750–4755, 2008. doi:10.1109/TIT.2008.928987.

26 Sofya Raskhodnikova, Dana Ron, Amir Shpilka, and Adam Smith. Strong lower bounds for approximating distribution support size and the distinct elements problem. *SIAM Journal on Computing (SICOMP)*, pages 813–842, 2009. doi:10.1137/070701649.

27 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing (SICOMP)*, pages 252–271, 1996. doi:10.1137/S0097539793255151.

28 Gregory Valiant and Paul Valiant. Estimating the unseen: improved estimators for entropy and other properties. *Journal of the ACM (JACM)*, pages 1–41, 2017. doi:10.1145/3125643.

29 Paul Valiant. Testing symmetric properties of distributions. *SIAM Journal on Computing (SICOMP)*, pages 1927–1968, 2011. doi:10.1137/080734066.

30 Yihong Wu and Pengkun Yang. Minimax rates of entropy estimation on large alphabets via best polynomial approximation. *IEEE Transactions on Information Theory*, pages 3702–3720, 2016. doi:10.1109/TIT.2016.2548468.

31 Yihong Wu and Pengkun Yang. Chebyshev polynomials, moment matching, and optimal estimation of the unseen. *The Annals of Statistics*, pages 857–883, 2019.

32 Yihong Wu, Pengkun Yang, et al. Polynomial methods in statistical inference: theory and practice. *Foundations and Trends® in Communications and Information Theory*, pages 402–586, 2020.

The Randomness Complexity of Differential Privacy

Clément L. Canonne
University of Sydney, Australia

Francis E. Su
Harvey Mudd College, Claremont, CA, USA

Salil P. Vadhan
Harvard University, Cambridge, MA, USA

Abstract

We initiate the study of the *randomness complexity* of differential privacy, i.e., how many random bits an algorithm needs in order to generate accurate differentially private releases. As a test case, we focus on the task of releasing the results of d counting queries, or equivalently all one-way marginals on a d-dimensional dataset with boolean attributes. While standard differentially private mechanisms for this task have randomness complexity that grows linearly with d, we show that, surprisingly, only $\log_2 d + O(1)$ random bits (in expectation) suffice to achieve an error that depends polynomially on d (and is independent of the size n of the dataset), and furthermore this is possible with pure, unbounded differential privacy and privacy-loss parameter $\varepsilon = 1/\operatorname{poly}(d)$. Conversely, we show that at least $\log_2 d - O(1)$ random bits are also necessary for nontrivial accuracy, even with approximate, bounded DP, provided the privacy-loss parameters satisfy $\varepsilon, \delta \leq 1/\operatorname{poly}(d)$. We obtain our results by establishing a close connection between the randomness complexity of differentially private mechanisms and the geometric notion of "deterministic rounding schemes" recently introduced and studied by Vander Woude et al. (2022, 2023).

2012 ACM Subject Classification Theory of computation → Pseudorandomness and derandomization; Theory of computation → Computational complexity and cryptography; Security and privacy; Mathematics of computing

Keywords and phrases differential privacy, randomness, geometry

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.27

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2024/169/`

Funding *Clément L. Canonne*: Supported by an ARC DECRA (DE230101329).
Francis E. Su: Work done in part while visiting and supported by the University of Sydney Mathematical Research Institute.
Salil P. Vadhan: Work done in part while visiting and supported by the University of Sydney Department of Computer Science and the Sydney Mathematical Research Institute. Also supported in part by Cooperative Agreement CB20ADR0160001 with the U.S. Census Bureau and a Simons Investigator Award.

Acknowledgements We thank Mark Bun for pointing out the connections to recent work on replicability, and anonymous referees for helpful corrections and comments.

1 Introduction

Differential privacy [5] is a widely accepted theoretical framework for protecting the privacy of individuals in a database while analysts query the database for statistical information. Differentially private (DP) mechanisms provide quantitative guarantees and tradeoffs on the level of privacy afforded to individuals and the accuracy of answers to queries. In order to provide these guarantees, DP mechanisms rely on the use of carefully calibrated "random noise" to protect privacy. Thus, large-scale deployments of differential privacy can require a


© Clément L. Canonne, Francis E. Su, and Salil P. Vadhan;
licensed under Creative Commons License CC-BY 4.0



massive amount of high-quality random (or cryptographically pseudorandom) bits.[1] Indeed, Garfinkel and LeClerc [7] estimate that the U.S. Census Bureau's differentially private TopDown Algorithm as used for the 2020 Decennial Census required at least 90 terabytes of random bits, and described major engineering challenges in generating those bits with sufficient efficiency and security. As they write in their conclusion,

> *"The need to generate a large number of high-quality random numbers is a largely unrecognized requirement of a production differential privacy system."*

Motivated by these challenges, and following the long line of inquiry in theoretical computer science on the role and necessity of randomness in computation, we initiate the study of the *randomness complexity of differential privacy*:

> What is the minimum amount of randomness required by differentially private mechanisms to achieve a specific level of accuracy?

We will quantify this "minimal amount of randomness" using either the maximum or expected *number of random bits* used by a differentially private algorithm, as a function of the dataset dimensions, privacy-loss parameters, and accuracy. For the sake of concreteness, we focus in this work on a specific task, *summation* (or *one-way marginal*) queries, which asks to provide an estimate of the sum of d-dimensional (binary) vectors, each corresponding to a different individual in the dataset.

In more detail, the task of summation is as follows: the dataset x consists of data from n individuals, each contributing a d-dimensional binary vector x_i. We can think of x as a $n \times d$ matrix with rows $x_1, ..., x_n$. The mechanism $M(x)$ must output an estimate $\hat{x} \in \mathbb{R}^d$ of $\text{sum}(x) := \sum_{i=1}^{n} x_i$, such that $\|\hat{x} - \text{sum}(x)\|_\infty \leq \alpha$ with probability at least $1 - \beta$. We call such a mechanism (α, β)-*accurate*. (See Definition 2.3 for a formal definition.)

We require that the mechanism M also satisfies (ε, δ)-*differential privacy* ((ε, δ)-*DP*) for some parameters $\varepsilon \geq 0, \delta \in [0, 1]$, which means that for every pair of datasets x, x' that differ on one row, we have

$$\forall S \qquad \Pr[M(x) \in S] \leq e^{\varepsilon} \cdot \Pr[M(x') \in S] + \delta.$$

When $\delta = 0$, this is called *pure* differential privacy, and we say that M is ε-*DP*. Our algorithm will work in the so-called *unbounded DP* setting, where the size n of the dataset is private information and thus needs to be protected by differential privacy like any other statistic. In contrast, our lower bound applies to the (easier) bounded DP setting, where n is public and fixed. (See Definitions 2.2 and 2.3 for formal definitions.)

We define the *maximum randomness complexity* $R_0(M)$ of a mechanism $M \colon (\{0,1\}^d)^n \to \mathbb{R}^d$ as the maximum over databases $x \in (\{0,1\}^d)^n$ of the maximum number of random bits used by M on input x. We define the *expected randomness complexity* $R(M)$ of a mechanism $M \colon (\{0,1\}^d)^n \to \mathbb{R}^d$ as the maximum over databases $x \in (\{0,1\}^d)^n$ of the expected number of random bits used by M on input x.

Differentially private summation is a well-studied question, for which many approximately optimal (in terms of accuracy) DP mechanisms have been proposed and analyzed. One of the simplest is the Laplace mechanism, which consists in adding d-dimensional Laplace noise

[1] In this work, we focus on using truly random bits to achieve the standard definition of DP with information-theoretic security. Later in the introduction, we discuss how this relates to the use of pseudorandom bits to implement DP with computational security, as typically done in practice.

with scale parameter d/ε to the true value $\mathrm{sum}(x)$. This can be shown to have ℓ_∞ error $\tilde{O}(d)/\varepsilon$ with high probability, which is optimal up to log factors. (Here $\tilde{O}(f)$ is shorthand for a function that is $O(f \cdot \log^c f)$ for an unspecified constant $c > 0$.) However, this comes at a significant cost, as adding continuous Laplace noise would technically require an infinite number of random bits. Instead, one can choose to add (independent) geometric noise to each coordinate of the true count, using a symmetrized Geometric random variable. This similarly will achieve nearly optimal ℓ_∞ error, but now with a finite expected randomness complexity. Unfortunately, the expected number of random bits required, $\tilde{O}_\varepsilon(d)$, still scales at least linearly with the dimension. (Here the $\tilde{O}_\varepsilon$ means that ε is treated as a constant, on which the hidden constants in the $\tilde{O}$ can depend.)

At first, this linear scaling seems inherent: For $d = 1$, to achieve non-trivial accuracy every DP mechanism must have entropy $\Omega(1)$:

▶ **Lemma 1.1.** *For every $\alpha < n/2$ and $\beta < 1/2$ and every (α, β)-accurate ε-DP mechanism $M\colon \{0,1\}^n \to \mathbb{R}$ for binary counts, there is some database $x \in \{0,1\}^n$ such that $H(M(x)) \geq 1 - \frac{\varepsilon}{2 \ln 2}$.*

Here $H(M(x))$ denotes the Shannon entropy of the output distribution $M(x)$, which is a lower bound on the expected number of random bits used by M. (We provide a simple proof of Lemma 1.1 in the full version.) Given that, in the d-dimensional summation task, the d dimensions are totally independent, one might expect the entropy required to be additive across dimensions, leading to an $\Omega(d)$ lower bound on expected randomness complexity.

Our results

Surprisingly, the above intuition turns out to be false: by leveraging recent work on *deterministic rounding schemes*, we show that the expected randomness complexity can be made as low as $\log_2 d + O(1)$, while still achieving good accuracy (i.e., an error that is polynomial in the dimension d, independent of the size n of the database):

▶ **Theorem 1.2** (Informal; see Corollary 3.2)**.** *For every $\ell > 0$ and $\varepsilon > 0$, there is an ε-DP mechanism $M\colon (\{0,1\}^d)^* \to \mathbb{R}^d$ (in the unbounded DP setting) such that the expected randomness complexity of M satisfies*

$$R(M) \leq \left\lceil \frac{d}{\ell} \right\rceil \cdot \log_2(\ell + 1) + O(1)$$

and for every $\beta \geq 1/\operatorname{poly}(d)$, M is (α, β)-accurate for summation with

$$\alpha = \frac{\tilde{O}(d) \cdot \ell}{\varepsilon}.$$

Taking $\ell = d$ gives expected randomness complexity $\log_2 d + O(1)$ as claimed. Furthermore, with this setting of ℓ and $\varepsilon \geq 1/\operatorname{poly}(d)$, the accuracy is $\alpha = \operatorname{poly}(d)$, independent of n. If we instead take $\ell = 1$, we get near-optimal accuracy $\alpha = \tilde{O}(d)/\varepsilon$, but then the expected randomness complexity becomes $d + O(1)$. Choosing $1 < \ell < d$ provides a tradeoff between these two extremes.

For bounding the *maximum* randomness complexity, we necessarily[2] relax to approximate DP:

[2] If M uses at most r random bits, then for every database x, the support of the distribution $M(x)$ is of size at most 2^r. However, a pure DP algorithm must have the same support on every input database. A summation mechanism with nontrivial accuracy on datasets of size n should at least distinguish the 2^d datasets where all n rows are the same, and thus must have $r \geq d$.

▶ **Theorem 1.3** (Informal; see Corollary 3.4). *For every $\ell > 0$, $\varepsilon > 0$, $\delta \in (0, 1/d)$, there is an (ε, δ)-DP mechanism $M : (\{0,1\}^d)^* \to \mathbb{R}^d$ (in the unbounded DP setting) such that the maximum randomness complexity of M satisfies*

$$R_0(M) \leq \left\lceil \frac{d}{\ell} \right\rceil \cdot \log_2(\ell + 1) + \log_2(1/\delta) + O(1)$$

and for every $\beta \geq 1/\operatorname{poly}(d)$, M is (α, β)-accurate for summation with

$$\alpha = O\left(\frac{\sqrt{d} \cdot \ell \cdot \log(1/\delta)}{\varepsilon} \right).$$

Again, the randomness complexity is minimized at $\ell = d$, which gives $R_0(M) \leq \log_2 d + \log_2(1/\delta) + O(1)$ and $\alpha = O(d^{3/2}) \cdot \log(1/\delta)/\varepsilon$, and the latter is again a factor of $\Theta(d \cdot \sqrt{\log(1/\delta)})$ larger than the error achievable with unlimited randomness. Typically, δ is taken to be cryptographically negligible, so $\delta \leq d^{-\omega(1)}$ and our bound on the maximum randomness complexity becomes $R_0(M) \leq (1 + o(1)) \cdot \log_2(1/\delta)$.

Next, we prove lower bounds showing that the randomness complexity we achieve is nearly optimal (in certain parameter regimes):

▶ **Theorem 1.4** (Informal; see Corollary 4.2). *Suppose that $M : (\{0,1\}^d)^n \to \mathbb{R}^d$ is an (ε, δ)-DP mechanism (in the bounded DP setting) that is (α, β)-accurate for summation, where $\alpha \leq n/2 - 1$. Then the maximum randomness complexity of M satisfies*

$$R_0(M) \geq \min\{d, \log_2(1/\delta)\}.$$

If in addition, $\beta \leq 1/d$, $\varepsilon \leq 1/d$, and $\delta \leq 1/O(d^2)$. Then both the maximum and expected randomness complexities of M satisfy:

$$R_0(M) \geq R(M) \geq \log_2 d - O(1).$$

Let's examine the constraint on the parameters. The constraint on α essentially just requires that M has nontrivial accuracy. Indeed, accuracy $n/2$ on datasets of size n can trivially be achieved by the deterministic mechanism that always outputs $(n/2, n/2, \ldots, n/2) \in \mathbb{R}^d$; this mechanism is $(n/2, 0)$-accurate, 0-DP, and has randomness complexity 0. The constraint on δ is quite mild, since δ is intended to be cryptographically small, and in particular subpolynomial in input size parameters like d and n. The constraints on β and ε are more nontrivial. They match our upper bound, in the sense that Theorem 1.2 achieves expected randomness complexity $\log_2 d + O(1)$ even when $\beta, \varepsilon = 1/d$. However, it would be interesting to know whether even smaller randomness complexity is possible when ε and β are constants.

Connection to geometry

The key ingredient in our results is a two-way connection to the notion of *deterministic rounding schemes* recently introduced and studied by Vander Woude, Dixon, Pavan, Radcliffe, and Vinodchandran [11, 12]. Deterministic rounding schemes provide methods for rounding data in $\mathbb{R}^d$ to nearby points so that any ball of small enough radius rounds to only a small number of points. As discussed in [11, 12], such rounding schemes are equivalent to the geometric notion of a *secluded partition*, which is a partition of $\mathbb{R}^d$ into sets of bounded radius such that balls of a sufficiently small radius do not intersect too many sets of the partition. We tie these ideas to the randomness complexity of accurate DP algorithms. In particular, our $\log_2 d \pm \Theta(1) = \log_2(d + 1) \pm \Theta(1)$ upper and lower bounds on randomness complexity

are intimately tied to the fact that in d dimensions, bounded-radius sets that cover $\mathbb{R}^d$ must contain intersections of at least size $d+1$ at certain points, and it is possible to find covers where no more than $d+1$ sets ever jointly intersect. In the theory of set intersections, these ideas are closely related to KKM covers of bounded polyhedra and the Polytopal Sperner lemma [3], and in fact our initial proofs of our results (not included here) came through these connections.

Related work on replicability

Recent work [4] has studied the randomness complexity of *replicable algorithms* using similar geometric tools. There are bidirectional conversions between replicable algorithms and approximate differentially private algorithms [8, 1] for problems about "population statistics," where the dataset consists of iid samples from an unknown distribution and the goal is to estimate statistics abut the distribution. In contrast, our focus is on "empirical statistics," where there is no iid assumption on the dataset and the goal is to estimate statistics of the dataset itself (as in the motivating use case of the 2020 U.S. Census). One can convert differentially private algorithms for population statistics into ones for empirical statistics (see, e.g., [2]), but this conversion incurs a high cost in randomness complexity (running the DP algorithm for population statistics on a dataset formed by sampling n rows from the input dataset independently with replacement). In the reverse direction, differentially private algorithms for empirical statistics are also automatically differentially private algorithms for population statistics (provided the dataset size n is large enough so that the empirical statistics approximate the population statistics with high probability), but converting a DP algorithm for population statistics into a replicable algorithm also appears to incur a large cost in randomness complexity. Thus neither our upper bound nor our lower bound appear to follow as a black box from the existing results on replicability, but it will be interesting to explore whether the techniques in either setting can be used to improve or extend any of the results in the other.

On using pseudorandom bits

In practice, implementations of differential privacy (including for the 2020 Decennial U.S. Census [7]) do not use truly random bits, but use pseudorandom bits generated by a cryptographically strong pseudorandom generator from a short seed, which is assumed to be truly random (or at least have sufficiently high entropy). If the differentially private algorithms satisfy the standard information-theoretic definition of DP (Definition 2.2) when run with truly random bits, then when executed using cryptographically strong pseudorandom bits, they satisfy a natural and convincing relaxation of differential privacy to computationally bounded adversaries [9]. Although the use of a pseudorandom generator reduces the number of truly random bits needed to equal the seed length of the generator, there is still a substantial cost if the differentially private algorithm is designed to use a huge number of random bits, since we must run the pseudorandom generator enough times to produce all of those bits. This is the cost referred to by Garfinkel and LeClerc [7], and it motivates our focus on the number of truly random bits needed to achieve information-theoretic DP. Furthermore, our Theorem 1.2 shows that we can reduce the expected number of truly random bits to be even smaller than the seed length of a cryptographic pseudorandom generator. Specifically, a cryptographic pseudorandom generator requires seed length at least $\log_2(1/\delta)$, where δ is a bound on the probability with which any computationally bounded adversary can distinguish the pseudorandom bits from uniform. This δ plays an analogous role in computational

differential privacy to the δ in information-theoretic (ε, δ)-DP. Typically, δ is taken to be cryptographically negligible, so $\delta = d^{-\omega(1)}$ and the randomness complexity $\log_2(d+1)+O(1)$ of Theorem 1.2 is asymptotically smaller than $\log_2(1/\delta)$.

Overview of the proofs

The high-level idea of our upper bound (algorithmic result) starts with the following observation: if we could post-process the output of a DP mechanism M to only allow a small number of outcomes on every database without hurting its accuracy too much, then we would be in good shape: for every database x, having $|\mathrm{supp}(M(x))| \leq k$ implies that $H(M(x))$ is at most $\log_2 k$, and so we obtain a new mechanism M' with low entropy but similar accuracy. Now, M' could still have very high randomness complexity $R(M')$ (since M itself could have been using a lot of random bits), but it is a standard fact in information theory (based on Huffman coding) that any discrete random variable Y can be generated using at most $H(Y)+O(1)$ random bits on average. Thus, there is yet another mechanism M'' with randomness complexity at most $\log_2 k + O(1)$.

Unfortunately, if we want to achieve pure DP as in Theorem 1.2, it is too much to hope for bounding the randomness complexity via support size. (See Footnote 2.) Instead, we will ensure that there is a set S_x of size at most k such that $M(x) \in S_x$ with probability at least $1-\gamma$, for a small $\gamma > 0$; we'll take $\gamma = 1/d$. If we can also ensure that the entropy of $M(x)$ conditioned on $M(x) \notin S_x$ is bounded by $O(1/\gamma)$, then this suffices to achieve an overall entropy of at most $\log_2 k + O(1)$.

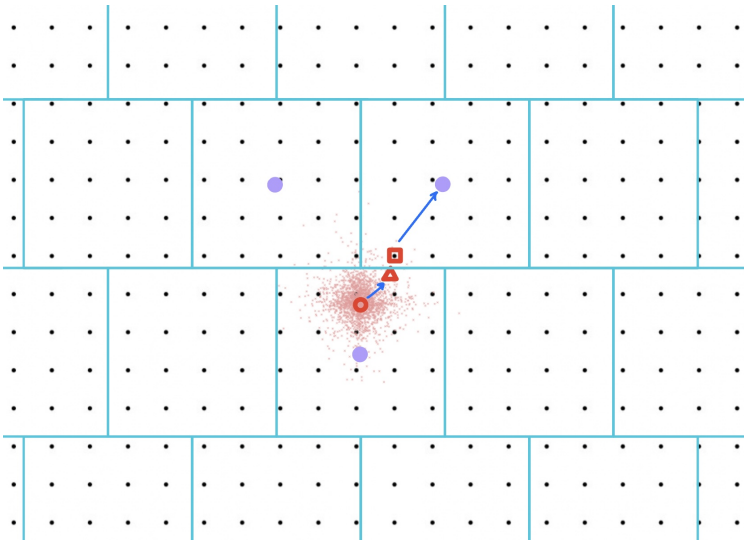


Figure 1 Using a deterministic rounding scheme (the square cells round to their center points) to construct a DP mechanism M for $\mathrm{sum}(x)$ with low randomness complexity. Given a dataset x with $\mathrm{sum}(x)$ at the annular dot, we add Laplace noise (depicted here by the diamond cloud of points). With high probability this lands somewhere nearby (the triangular dot). Next, we round that to the nearest point in a suitably chosen grid (the square dot) – the limited grid options help keep the resulting entropy low. We then use the given deterministic rounding scheme to jump to the center of the cell containing the square dot. This defines a DP mechanism with low entropy, because there are only a small number k (here, $k = 3$) of cell centers that could be in the high probability support of this mechanism applied to x.

To implement the above plan, we start with any accurate ε-DP mechanism M, say one based on the Laplace mechanism. The crucial step is then how to achieve the post-processing step. This we do with a deterministic rounding scheme f. Such a scheme will take the output $M(x)$ and round it to a nearby point $f(M(x))$, say at ℓ_∞ distance at most r from $M(x)$. We choose r to be large enough so that the noise in $M(x)$ will be of magnitude (in ℓ_∞ norm) at most $r \cdot \tau$ with high probability, for a sufficiently small τ. f is called a (k, τ) deterministic rounding scheme of radius r if on every ℓ_∞ ball of radius $r \cdot \tau$, f takes on at most k distinct values. With such a scheme, we get that $f(M(x))$ lies in a set of size k with high probability.

So then the question becomes what parameters (k, τ) are possible, and this is exactly what is addressed in the work of Vander Woude et al. [12]. In particular, they construct a scheme with $k = (d+1)^\ell$ and $\tau = 1/(2\ell)$, which is what we use in Theorem 1.2. As noted above, completing the proof of the theorem requires controlling the entropy of $M(x)$ also conditioned on the event that the noise is large. This we achieve by an additional coarsening of the output of $M(x)$, via a standard rounding of every coordinate to a multiple of $r \cdot \tau/2$, before applying the deterministic rounding scheme f.

For our upper bound on maximum randomness complexity (Theorem 1.3), we apply the same strategy as above to the Gaussian mechanism for differential privacy. In this case, we replace our use of Huffman coding with the fact that if we have a random variable Y that lies in a set S of size at most k with probability at least $1 - \gamma$, then for every $\eta > 0$, using at most $\log_2 k + \log_2(1/\eta) + O(1)$ random bits, we can generate a random variable Y' that is at total variation distance at most $\eta + \gamma$ from Y.

To prove our lower bound of $\log_2 d - O(1)$ (second part of Theorem 1.4) on expected randomness complexity, we show how randomness-efficient DP mechanisms for summation with good accuracy imply the existence of deterministic rounding schemes with good parameters: where the parameters k, τ of the rounding schemes are directly related to the randomness complexity of the DP mechanism M. This allows us to invoke impossibility results from [11, 12] on the existence of "too-good-to-be-true" deterministic rounding schemes to rule out DP mechanisms which are simultaneously accurate and randomness-efficient. In more detail, the main steps of the lower bound are as follows: given a purported ε-DP mechanism M for summation with low randomness complexity, (1) we embed the hypergrid $[n]^d$ into the space of d-dimensional datasets of size n, $(\{0,1\}^d)^n$, such that ℓ_1 distance in the former maps to Hamming distance in the latter and computing summation on a given dataset $x = x^{(v)}$ allows one to retrieve the original hypergrid vector $v \in [n]^d$. (2) We use the randomness guarantees of M to extract, from its output distribution on a given database $x^{(v)}$, the highest-probability representative output $y^{(v)}$, which defines a deterministic rounding scheme f:

$$\underbrace{v \in [n]^d \rightsquigarrow x^{(v)} \overset{M}{\rightsquigarrow} y^{(v)} \in \mathbb{R}^d}_{f}$$

(3) we leverage the accuracy guarantees of M to argue that this rounding y is indeed close to v; and, finally, (4) we invoke the (group) privacy guarantee of M to show that the image of any given ℓ_∞ ball by our newly-defined rounding scheme cannot contain too many "representatives" $y^{(v)}$. There is one last step required, as the rounding scheme f outline above is only defined on $[n]^d$: to conclude, we need to "lift" this rounding scheme to the whole of $\mathbb{R}^d$ while preserving its properties, which we do by a careful tiling of the space $\mathbb{R}^d$ with translations and reflections of the hypergrid.

The lower bound of $\min\{d, \log_2(1/\delta)\}$ on maximum randomness complexity follows from observing that any (ε, δ)-DP mechanism that uses less than $\log_2(1/\delta)$ random bits is ε'-DP for some finite ε', and then we can apply the argument of Footnote 2 to deduce $R_0(M) \geq d$.

Directions for Future Work

We conclude this introduction by mentioning some open questions, and directions for future work.

The first question is whether one can close the gap between our upper and lower bounds. Our upper bounds show that randomness complexity $\log_2 d + O(1)$ can be achieved with accuracy $\alpha = \text{poly}(d)/\varepsilon$, but to achieve near-optimal accuracy $\alpha = \tilde{O}(d)/\varepsilon$, we only achieve randomness complexity $d + O(1)$. Can our algorithm be improved to achieve logarithmic randomness complexity with near-optimal accuracy? Or can our lower bound be improved to show that linear randomness complexity is necessary for near-optimal accuracy? As discussed earlier, another question is to remove some of the constraints on parameters, like ε and β, in our lower bound, or else show that sub-logarithmic randomness complexity is possible when these parameters are constant.

A second question whether one can obtain *efficient* low-randomness DP mechanisms, e.g., running in time $\text{poly}(n, d)$. Recall that our mechanisms from Theorem 1.2 and Theorem 1.3 rely on computationally inefficient procedures (e.g., Huffman coding) to convert a mechanism with low output entropy into one with low randomness complexity. For this reason, as well as the aforementioned accuracy loss, we would not recommend using our mechanisms in practice (even though our work was inspired by the practical concerns raised in [7]). Still, it raises the hope for practical DP mechanisms that are much more randomness-efficient than ones currently used.

As a test case we focused in this paper on the task of differentially private summation: extending our study of the randomness requirement of DP mechanisms to other statistical releases, or attempting to provide a general treatment of the randomness complexity of differential privacy, would be an interesting direction.

Finally, as mentioned earlier, exploring connections to recent work on replicable algorithms may also prove to be fruitful.

2 Preliminaries

2.1 Differential Privacy

Consider a database x consisting of n entries chosen from some domain $\mathcal{X}$: that is, x is an element of $\mathcal{X}^n$. If each entry of the database consists of d numerical attributes, then $\mathcal{X}$ might be $\mathbb{R}^d$ or $\mathbb{N}^d$ (for discrete data) or $\{0,1\}^d$ (for binary data). It is convenient to think of x as an $n \times d$ matrix, with each row of the matrix being the vector $x_i = (x_{ij})_{1 \le j \le d}$. In general, when the size of the database n is unknown, or allowed to grow, we will denote it by $|x|$, and accordingly consider databases in $\mathcal{X}^* = \cup_{n=0}^{\infty} \mathcal{X}^n$.

▶ **Definition 2.1** (Database metrics and adjacency). *For two databases $x, x' \in \mathcal{X}^*$, their* insert-delete distance *(aka LCS distance) $D_{\text{ID}}(x, x')$ is the number of insertions and deletions of elements of $\mathcal{X}$ to transform x into x'. For two databases $x, x' \in \mathcal{X}^*$ such that $|x| = |x'|$, their* Hamming distance *$D_{\text{Ham}}(x, x')$ is the number of rows i such that $x_i \neq x'_i$. If $|x| \neq |x'|$, we define $D_{\text{Ham}}(x, x') = \infty$. For a database metric D, we say that x and x' are* adjacent *with respect to D, denoted $x \sim_D x'$ if $D(x, x') \le 1$.*

We use D_{ID} to capture *unbounded differential privacy*, where the size n of the dataset is unknown and considered private information that needs to be protected. D_{Ham} captures *bounded differential privacy*, where the size n is public information, known to a potential adversary. Since $D_{\text{ID}}(x, x') \le 2 \cdot D_{\text{Ham}}(x, x')$ for databases x, x' of the same size, unbounded-DP algorithms are also bounded DP, up to a factor of 2 in the privacy-loss parameters. We state our positive result in the unbounded-DP setting and our negative result in the bounded-DP setting, making both results stronger.

▶ **Definition 2.2** (Differential privacy). *Fix $\varepsilon > 0$ and $\delta \in [0,1]$. A randomized algorithm $M\colon \mathcal{X}^* \to \mathcal{Y}$ is* (ε,δ)-differentially private *(or (ε,δ)-DP) with respect to database metric D if for every pair of adjacent databases $x \sim_D x'$ in $\mathcal{X}^*$ and every measurable $S \subseteq \mathcal{Y}$, we have*

$$\Pr[M(x) \in S] \leq e^{\varepsilon} \cdot \Pr[M(x') \in S] + \delta.$$

If $\delta = 0$, we simply say M is ε-DP.

We restrict our attention in this article to databases where each record has d binary attributes, so $\mathcal{X} = \{0,1\}^d$, and to mechanisms that output an estimate of the sum of the attributes. This motivates the following definition:

▶ **Definition 2.3** (Accuracy). *Given $\alpha \geq 0$ and $\beta \in [0,1]$, a randomized algorithm $M\colon (\{0,1\}^d)^n \to \mathbb{R}^d$ is said to be* (α,β)-accurate for summation *if for every $x \in (\{0,1\}^d)^*$, we have*

$$\Pr[\|M(x) - \mathrm{sum}(x)\|_\infty > \alpha] \leq \beta,$$

where

$$\mathrm{sum}(x) = \sum_{i=1}^{|x|} x_i \in \mathbb{R}^d.$$

Note that we used $(\{0,1\}^d)^n$ rather than $(\{0,1\}^d)^*$ as the domain of M. This allows for the possibility that, even in the unbounded DP setting, the accuracy of the mechanism depends on the size n of the dataset, as well as the dimension d and the privacy parameters ε and δ.

Two key features of differential privacy are its *immunity to post-processing* and its *group privacy* property:

▶ **Lemma 2.4** (Postprocessing). *Let $M\colon \mathcal{X} \to \mathcal{Y}$ be an (ε,δ)-DP mechanism, and $f\colon \mathcal{Y} \to \mathcal{Z}$ be any (possibly randomized) function. Then $f \circ M$ is (ε,δ)-DP.*

▶ **Lemma 2.5** (Group privacy). *Let $M\colon \mathcal{X} \to \mathcal{Y}$ be an (ε,δ)-DP mechanism with respect to database metric D, and $x, x' \in \mathcal{X}^*$ be two databases at distance $D(x,x') \leq k$. Then, for every measurable $S \subseteq \mathcal{Y}$, we have*

$$\Pr[M(x) \in S] \leq e^{k\varepsilon} \cdot \Pr[M(x') \in S] + k e^{(k-1)\varepsilon}\delta.$$

We refer the reader to, e.g., [6, 10] for more background on differential privacy and the proof of these properties. We also briefly recall the definition and guarantees of two of the standard noise mechanisms used in differential privacy, the Laplace and Gaussian mechanisms:

▶ **Lemma 2.6** (Laplace mechanism; see e.g., [6, Theorems 3.6 and 3.8]). *Suppose $f\colon \mathcal{X}^* \to \mathbb{R}^d$ has ℓ_1 sensitivity $\Delta_1(f)$, that is, $\Delta_1(f) = \max_{x \sim_D x'} \|f(x) - f(x')\|_1$. Then, for every $\varepsilon > 0$, the mechanism $M\colon \mathcal{X}^* \to \mathbb{R}^d$ defined by*

$$M(x) = f(x) + \mathrm{Lap}(\Delta_1(f)/\varepsilon)^d$$

is ε-DP, where $\mathrm{Lap}(b)^d$ denotes the product distribution over $\mathbb{R}^d$ with iid marginals distributed as a Laplace with scale parameter $b > 0$ (i.e., probability density function $f(x) = \frac{1}{2b}e^{-|x|/b}$). Moreover, its accuracy satisfies the following: for every $\beta \in (0,1]$,

$$\Pr\left[\|M(x) - f(x)\|_\infty \geq \frac{\Delta_1(f)}{\varepsilon} \ln \frac{d}{\beta}\right] \leq \beta\,.$$

▶ **Lemma 2.7** (Gaussian mechanism; see e.g., [6, Theorems 3.22 and A.1]). *Suppose $f\colon \mathcal{X}^* \to \mathbb{R}^d$ has ℓ_2 sensitivity $\Delta_2(f)$, that is, $\Delta_2(f) = \max_{x \sim_D x'} \|f(x) - f(x')\|_2$. Then, for every $\varepsilon \in (0,1]$ and $\delta \in (0,1]$, the mechanism $M\colon \mathcal{X}^* \to \mathbb{R}^d$ defined by*

$$M(x) = f(x) + \mathcal{N}\left(0, \left(\frac{\Delta_2(f)}{\varepsilon} \cdot \sqrt{2 \ln \frac{1.25}{\delta}}\right)^2\right)^d$$

is (ε, δ)-DP, where $\mathcal{N}(0, \sigma^2)^d$ denotes the product distribution over $\mathbb{R}^d$ with iid marginals distributed as a Normal distribution with mean 0 and variance σ^2. Moreover, its accuracy satisfies the following: for every $\beta \in (0,1]$,

$$\Pr\left[\|M(x) - f(x)\|_\infty \geq \frac{\Delta_2(f)}{\varepsilon} \cdot 2\sqrt{\ln \frac{1.25}{\delta} \cdot \ln \frac{2d}{\beta}}\right] \leq \beta\,.$$

2.2 Randomness Complexity

We work with a model of randomized algorithms where the algorithm has access to a coin-tossing oracle that returns an independent, unbiased random bit on each invocation. The *number of random bits* used by M on a particular execution is defined to be the number of the calls to the coin-tossing oracle (which is a random variable, as the algorithm could adaptively decide whether to call the oracle again or not depending on the results of prior coin tosses). We require that on all inputs, the algorithm halts with probability 1 over the coin tosses received.

We define below two natural measures of randomness complexity. Like with accuracy, we will use $\mathcal{X}^n$ as the domain rather than $\mathcal{X}^*$ to allow the possibility that the randomness complexity depends on the size n of the dataset.

▶ **Definition 2.8** (randomness complexity). *For a randomized algorithm $M\colon \mathcal{X}^n \to \mathcal{Y}$ and $\gamma \in [0,1]$, we define:*

- *$R(M)$ = the maximum over $x \in \mathcal{X}^n$ of the expected number of random bits used by M on input x, where the expectation is taken over the coin tosses of M.*
- *$R_0(M)$: the maximum over x of the maximum number of random bits used by M on input x.*

Rather than work directly with randomness complexity, it is more convenient for us to work with measures of *output entropy*:

▶ **Definition 2.9.** *For a randomized algorithm $M\colon \mathcal{X}^n \to \mathcal{Y}$, we define*

- *$H(M) = \max_x H(M(x))$, where $H(Y)$ denotes the Shannon entropy of random variable Y (in bits), defined as $\mathbb{E}_{y \leftarrow Y}[\log_2(1/\Pr[Y=y])]$.*
- *$H_\infty(M) = \max_x H_\infty(M(x))$, where $H_\infty(Y)$ denotes the min-entropy of random variable Y, defined as $\log_2[1/(\max_y \Pr[Y=y])]$.*
- *$H_0(M) = \max_x H_0(M(x))$, where $H_0(Y)$ denotes the max-entropy of random variable Y, defined as $\log_2 |\mathrm{supp}(Y)|$.*
- *$H_0^\gamma(M) = \max_x H_0^\gamma(M(x))$, where $H_0^\gamma(Y)$ denotes the γ-smoothed max-entropy of random variable Y, which is defined to be the minimum of $H_0(Y \mid E)$ over (probabilistic) events $E = E(Y)$ of probability at least $1-\gamma$.*

Some basic relations between the randomness complexity measures and the output entropy measures are as follows:

▶ **Lemma 2.10.** *For any randomized algorithm $M\colon \mathcal{X}^n \to \mathcal{Y}$, the following hold:*

1. $H_\infty(M) \le H(M) \le H_0(M)$, $H_0^\gamma(M) \le H_0(M)$, $R(M) \le R_0(M)$.
2. $H(M) \le R(M)$ *and* $H_0(M) \le R_0(M)$.
3. *For every M, there is an M' such that $R(M') \le H(M) + O(1)$ and on every input x, $M'(x)$ is identically distributed to x'.*
4. *For every M and $\gamma, \eta > 0$, there is an M' such that $R_0(M') \le \lceil H_0^\gamma(M) + \log_2(1/\eta)\rceil$ and for every x, $M'(x)$ is at total variation distance at most $\gamma + \eta$ from $M(x)$.*

In particular, Item 2 says that a lower bound on output entropy is also a lower bound on randomness complexity. Items 3 and 4 say that we can go in the other direction as well, by modifying the mechanism.

2.3 Deterministic Rounding Schemes and Secluded Partitions

A crucial building block for our results is the notion of *deterministic rounding scheme*, recently introduced by Vander Woude, Dixon, Pavan, Radcliffe, and Vinodchandran [12]. Although they defined such a scheme as a function $f : \mathbb{R}^d \to \mathbb{R}^d$, we generalize their definition slightly to include functions $f : S \to \mathbb{R}^d$ on an arbitrary domain S and a radius parameter ρ that becomes necessary when S is not all of $\mathbb{R}^d$:

▶ **Definition 2.11.** *For $k \in \mathbb{N}$, $\rho, \tau \in \mathbb{R}^{\ge 0}$, and $S \subseteq \mathbb{R}^d$, a function $f : S \to \mathbb{R}^d$ is said to be a (k, τ)*-deterministic rounding scheme of radius ρ on S *if the following two conditions hold:*

1. *for all $z \in S$, $\|f(z) - z\|_\infty \le \rho$;*
2. *for all $z \in \mathbb{R}^d$, $|\{\, f(y) : y \in B_\infty(z, 2\tau \cdot \rho) \cap S \,\}| \le k$.*

where $B_\infty(z, r)$ denotes the closed ℓ_∞ ball of radius r centered at z. That is, each f "rounds" inputs to nearby points (within ρ), and inputs that are close (in a $2\tau \cdot \rho$-ball) can only be rounded to a small number k of representatives.

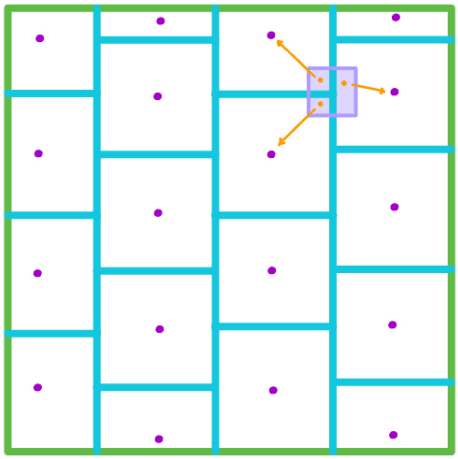


Figure 2 A (k, τ)-deterministic rounding scheme of radius $\rho \approx 0.14$ on $S = [0, 1]^2$, where $k = 3$ and $\tau = \frac{1}{4}$. Points in each cell round to one point inside that cell, never moving more than ρ in ℓ_∞-distance. The shaded box is a ℓ_∞ ball of radius $2\tau \cdot \rho$, and all balls of this size will round to at most $k = 3$ points.

When $S = \mathbb{R}^d$, the parameter ρ is not important; if f is a (k, τ) deterministic rounding scheme of radius ρ, then for every $c > 0$, we see that $f'(x) = cf(x/c)$ is a (k, τ) deterministic rounding scheme of radius $c\rho$. When ρ is not specified, its default value is $\rho = 1/2$, matching

the radius of "round-to-the-nearest-integer." However, we will also consider deterministic rounding schemes on $S = [0,1]^d$; then the choice of ρ is more important. Indeed, $[0,1]^d$ has a trivial $(1,\infty)$-deterministic rounding scheme of radius $1/2$, where all points get rounded to the center of $[0,1]^d$, but it has no $(1,\infty)$-deterministic rounding scheme of radius $1/4$.

As pointed out by Vander Woude et al. [12, Observation 1.3], deterministic rounding schemes have a nice geometric interpretation as (k,τ)-*secluded partitions*: partitions of S by sets of ℓ_∞ radius at most ρ such that every ℓ_∞ ball of radius $2\tau \cdot \rho$ intersects at most k sets in the partition. Using this geometric perspective, Vander Woude et al. prove the following upper and lower bounds:

▶ **Theorem 2.12** ([11, 12]). *For every $d \in \mathbb{N}$, there exists a $(d+1, \frac{1}{2d})$-deterministic rounding scheme $f : \mathbb{R}^d \to \mathbb{R}^d$. More generally, for every $\ell \in \mathbb{N}$, there exists a $((\ell+1)^{\lceil d/\ell \rceil}, \frac{1}{2\ell})$-deterministic rounding scheme $f : \mathbb{R}^d \to \mathbb{R}^d$.*

▶ **Theorem 2.13** ([11, 12]). *If $f : \mathbb{R}^d \to \mathbb{R}^d$ is a (k,τ)-deterministic rounding scheme and $\tau > 0$, then*

$$k \geq \max\{d+1, (1+2\tau)^d\}.$$

3 Upper bounds for summation

In this section, we establish our upper bound results, which state that good deterministic rounding schemes imply differentially private mechanisms for summation with low randomness complexity (Theorem 3.1).

We begin with our upper bound on expected randomness complexity.

▶ **Theorem 3.1.** *Suppose there exists a (k,τ)-deterministic rounding scheme $f : \mathbb{R}^d \to \mathbb{R}^d$. Then for every $\varepsilon > 0$, there is a mechanism $M : (\{0,1\}^d)^* \to \mathbb{R}^d$ that is ε-DP with respect to D_{ID} satisfying*

$$\begin{aligned} H_0^{1/d}(M) &\leq \log_2 k \qquad \text{and} \\ H(M) &\leq \log_2 k + O(1), \end{aligned}$$

such that, for every $\beta \in (0,1]$, M is (α,β)-accurate for summation for

$$\alpha = O\left(\frac{d \cdot \log(d/\beta)}{\varepsilon} + \frac{d \cdot \log d}{\tau\varepsilon}\right).$$

Combining this with Theorem 2.12 and the conversion from output entropy to randomness complexity (Lemma 2.10, Item 3), we get:

▶ **Corollary 3.2.** *For every $d, \ell, \varepsilon > 0$, there is a mechanism $M : (\{0,1\}^d)^* \to \mathbb{R}^d$ that is ε-DP with respect to D_{ID} satisfying*

$$R(M) \leq \left\lceil \frac{d}{\ell} \right\rceil \cdot \log_2(\ell+1) + O(1),$$

such that, for every $\beta \in (0,1]$, M is (α,β)-accurate for summation for

$$\alpha = O\left(\frac{d \cdot \log(d/\beta)}{\varepsilon} + \frac{\ell \cdot d \cdot \log d}{\varepsilon}\right).$$

In particular, taking $\ell = d$, we obtain $H(M) \leq \log_2 d + O(1)$ and

$$\alpha = O\left(\frac{d \cdot \log(d/\beta)}{\varepsilon} + \frac{d^2 \cdot \log d}{\varepsilon}\right).$$

Proof of Theorem 3.1. Let $f\colon \mathbb{R}^d \to \mathbb{R}^d$ be a (k,τ)-deterministic rounding scheme. Without loss of generality, by scaling, we can assume that f has radius $r/2$, for a parameter $r > 0$ to be determined later in the proof. Then we have that for every $z \in \mathbb{R}^d$,

1. $\|f(z) - z\|_\infty \le r/2$, and
2. $|\{\ f(y) :\ y \in B_\infty(z, r\cdot\tau)\ \}| \le k$.

Now define this mechanism M on a database x:

Mechanism $M(x)$.

1. Draw $\eta \sim \mathrm{Lap}(d/\varepsilon)^d$, and let $y = \mathrm{round}_{r\cdot\tau}(\mathrm{sum}(x) + \eta)$,[3]
2. Output $f(y)$.

By the post-processing property of differential privacy and the ε-DP guarantees of the Laplace mechanism, M is itself ε-DP: it remains to argue about its accuracy and randomness complexity. By the accuracy of the Laplace mechanism (Lemma 2.6), we have that, with probability at least $1-\beta$, the noise infusion introduces an error of at most

$$\|\eta\|_\infty \le \frac{\ln(d/\beta)\cdot d}{\varepsilon}.$$

Rounding the coordinates of $\mathrm{sum}(x) + \eta$ to the nearest multiple of $r\cdot\tau$ and replacing y with $f(y)$ increase the ℓ_∞ error by at most $r\cdot\tau/2 + r/2$, so we have (α,β)-accuracy for

$$\alpha = (1+\tau)\cdot\frac{r}{2} + \frac{\ln(d/\beta)\cdot d}{\varepsilon}.$$

For the randomness complexity, by the same analysis as in the accuracy above (but replacing β with $1/d$), we have that with probability at least $1 - 1/d$, the following event E holds:

$$\text{event } E : \|\eta\|_\infty \le r_0 \text{ for } r_0 := \frac{2d\ln d}{\varepsilon}.$$

Conditioned on E, the point y lies in an ℓ_∞ ball of radius $r_0 + r\cdot\tau/2$ around $\mathrm{sum}(x)$. By setting $r = 2r_0/\tau$, y lies in a ball of radius $2r_0 = r\cdot\tau$, and the fact that f is a (k,τ) deterministic rounding scheme of radius r tell us that, conditioned on E, the support size of $M(x)$ is at most k, and in particular has entropy at most $\log_2 k$. Thus, we have $H_0^{1/d}(M(x)) \le \log_2 k$.

To bound $H(M(x))$ overall, we also need to bound the entropy of $M(x)$ conditioned on $\neg E$, which is upper bounded by the entropy of $y = \mathrm{round}_{r\cdot\tau}(\mathrm{sum}(x) + \eta)$ conditioned on $\neg E$. $\neg E$ is the event that at least one coordinate of η has magnitude larger than r_0; the remainder may or may not have magnitude larger than r_0. Conditional on $\neg E$, coordinate i of y is distributed as $\mathrm{round}_{r\cdot\tau}(\mathrm{sum}(x)_i + \eta_i)$, where η_i is a mixture of $\mathrm{Lap}(d/\varepsilon)$ conditioned on having magnitude at most r_0 and $\mathrm{Lap}(d/\varepsilon)$ conditioned on having magnitude greater than r_0. Provided that $r\cdot\tau/2 = \Omega(d/\varepsilon)$, such a distribution has entropy $O(1)$, similarly to how a geometric distribution with parameter $p = \Omega(1)$ has entropy $H(p)/p = O(1)$.[4] By

[3] Note that the distribution of y is *not* the same as applying the Geometric Mechanism (i.e. Discrete Laplace Mechanism) supported on integer multiples of $r\cdot\tau$. To apply the Geometric Mechanism, we would need to first round $\mathrm{sum}(x)$ to a multiple of $r\cdot\tau$, which would substantially increase the sensitivity and increase the amount of noise we need to add. We could apply the Geometric Mechanism to $\mathrm{sum}(x)$ instead of the Laplace Mechanism prior to the rounding step, but it does not simplify anything in the analysis below.

[4] For a bit more detail: observe that, before conditioning, the distribution of $\mathrm{round}_{r\cdot\tau}(\mathrm{sum}(x)_i + \eta_i)$ is a mixture of a point mass and two geometric distributions: a point mass on $m = \mathrm{round}_{r\cdot\tau}(\mathrm{sum}(x)_i)$, a geometric distribution on the multiples of $r\cdot\tau/2$ smaller than m (assuming wlog that $m < \mathrm{sum}(x)_i$), and a geometric distribution supported on the multiples of $r\cdot\tau/2$ greater than or equal to $\mathrm{sum}(x)_i$.

our setting of r above, $r \cdot \tau/2$ is larger than d/ε by a factor of $2\ln(d) = \Omega(1)$. With $O(1)$ entropy contributed from each coordinate, we get $O(d)$ entropy overall conditioned on $\neg E$. (The coordinates of y are not independent conditioned on $\neg E$, but entropy is subadditive even for dependent random variables.)

Thus, letting I be the indicator random variable for event E, we have

$$\begin{aligned} H(M(x)) &\leq H(M(x), I) \\ &= H(M(x) \mid I) + H(I) \\ &\leq \Pr[E] \cdot H(M(x) \mid E) + \Pr[\neg E] \cdot H(M(x) \mid \neg E) + H\left(\frac{1}{d}\right) \\ &\leq 1 \cdot \log_2 k + \frac{1}{d} \cdot O(d) + 1 \\ &= \log_2 k + O(1). \end{aligned}$$

The accuracy bound follows by plugging our setting for r_0 and $r = 2r_0/\tau$ into the expression for α above. ◀

Now we turn to our upper bound on maximum randomness complexity.

▶ **Theorem 3.3.** *Suppose there exists a (k, τ)-deterministic rounding scheme $f\colon \mathbb{R}^d \to \mathbb{R}^d$. Then for every $\varepsilon > 0$, there is a mechanism $M\colon (\{0,1\}^d)^* \to \mathbb{R}^d$ that is (ε, δ)-DP with respect to D_{ID} satisfying $H_0^\gamma(M) \leq \log_2 k$ and such that, for every $\beta \in (0, 1]$, M is (α, β)-accurate for summation for*

$$\alpha = O\left(\frac{\sqrt{d \cdot \log(1/\delta) \cdot \log(d/\beta)}}{\varepsilon} + \frac{\sqrt{d \cdot \log(1/\delta) \cdot \log(d/\gamma)}}{\tau\varepsilon}\right).$$

Combining this with Theorem 2.12 and the conversion from output entropy to randomness complexity (Lemma 2.10, Item 4), we get:

▶ **Corollary 3.4.** *For every $d, \ell \in \mathbb{N}$, $\varepsilon > 0$ and $\delta \in (0, 1/d)$, there is a mechanism $M\colon (\{0,1\}^d)^* \to \mathbb{R}^d$ that is (ε, δ)-DP with respect to D_{ID} satisfying*

$$R_0(M) \leq \left\lceil \frac{d}{\ell} \right\rceil \cdot \log_2(\ell + 1) + \log_2\left(\frac{1}{\delta}\right) + O(1),$$

such that, for every $\beta \in (0, 1]$, M is (α, β)-accurate for summation for

$$\alpha = O\left(\frac{\sqrt{d \cdot \log(1/\delta) \cdot \log(d/\beta)}}{\varepsilon} + \frac{\ell \cdot \sqrt{d} \cdot \log(1/\delta)}{\varepsilon}\right).$$

In particular, taking $\ell = d$, we obtain $R_0(M) \leq \log_2 d + \log_2(1/\delta) + O(1)$ and

$$\alpha = O\left(\frac{\sqrt{d \cdot \log(1/\delta) \cdot \log(d/\beta)}}{\varepsilon} + \frac{d^{3/2} \cdot \log(1/\delta)}{\varepsilon}\right).$$

Proof Sketch of Theorem 3.3. Follow the proof of Theorem 3.1 but replace the use of the Laplace mechanism (Lemma 2.6) with the Gaussian mechanism (Lemma 2.7), and define the event E using the accuracy bound r_0 that holds with probability at least $1 - \gamma$ (rather than $1 - 1/d$). In this proof, there is no need to analyze $H(M(x)|\neg E)$. ◀

Under the assumption that $r \cdot \tau = \Omega(d/\varepsilon)$, the parameter p of these geometric distributions is bounded away from 1, i.e., the probability mass of each point in the support is a constant factor smaller than the point in the support closer to m. This latter property is preserved under conditioning on $|\eta_i| \leq r_0$ or conditioning on $|\eta_i| > r_0$, and suffices to ensure entropy $O(1)$.

4 Lower bound for summation

We now turn to establishing a lower bound on the randomness complexity of any accurate DP mechanism for summation. To strengthen the result, we will consider the bounded DP setting and allow approximate DP mechanisms; and our conclusion will yield a lower bound on the min-entropy $H_\infty(M)$ of any such DP mechanism.

▶ **Theorem 4.1.** *Suppose that $M\colon (\{0,1\}^d)^n \to \mathbb{R}^d$ is a mechanism that is a (ε,δ)-DP mechanism with respect to D_{Ham} with $H_\infty(M) \le \log_2 K$ that is (α,β)-accurate for summation for $\beta < 1/K$ and $\alpha \le n/2-1$. Then, for every $\tau > 0$, there exists a (k,τ)-deterministic rounding scheme with*

$$k \le \frac{K \cdot e^{\varepsilon h}}{1 - hKe^{\varepsilon h}\delta}$$

where

$$h = \min\left(d \cdot \left(8(\alpha+1)\cdot\frac{\tau}{1-2\tau}+1\right), n\right).$$

Observe that when we take $\tau \to 0$, we have $h \to \min\{d,n\} \le d$, so, at least when $\delta = 0$ our upper bound on k becomes $k \le K \cdot e^{\varepsilon d}$. By the lower bound on deterministic rounding schemes (Theorem 2.13), we know that $k \ge d+1$, so we obtain an entropy lower bound of $H_\infty(M) = \log_2 K \ge \log_2(d+1) - O(\varepsilon d)$. When $\varepsilon = O(1/d)$, this matches our best upper bound on randomness complexity up to an additive constant (the case $\ell = d$ in Corollary 3.2). For positive δ, we only lose an additive constant in the lower bound provided that $\delta \le 1/(2dKe^{\varepsilon d})$. Using the fact that $H_\infty(M) \le H(M) \le R(M)$, we obtain:

▶ **Corollary 4.2.** *Suppose that $M\colon (\{0,1\}^d)^n \to \mathbb{R}^d$ is a mechanism that is (ε,δ)-DP mechanism with respect to D_{Ham} with $\alpha \le n/2-1$, $\varepsilon \le 1/d$, $\beta < 1/d$, and $\delta \le 1/(6d^2)$. Then*

$$R(M) \ge \log_2 d - O(1).$$

(The conditions involving K in Theorem 4.1 disappear by doing a case analysis. If $H_\infty(M) > \log_2 d$, then we are done. Otherwise, we can set $K = d$ in Theorem 4.1.)

We will prove Theorem 4.1 by first constructing a deterministic rounding scheme for the cube $[0,1]^d$ and then lifting it to a deterministic rounding scheme for all of $\mathbb{R}^d$ by the following lemma (whose proof we defer to later).

▶ **Lemma 4.3.** *Suppose there exists a (k,τ)-deterministic rounding scheme $f\colon [0,1]^d \to \mathbb{R}^d$ of radius $\rho < 1/2$, where $\tau \in (0,1)$. Then there exists a (k,ζ)-deterministic rounding scheme $\tilde{f}\colon \mathbb{R}^d \to \mathbb{R}^d$, for $\zeta = \max\{\frac{\tau}{4-2\tau}, \frac{\tau}{2+4\rho}\}$.*

Proof of Theorem 4.1. By Lemma 4.3, it suffices to to construct a (k,τ')-deterministic rounding scheme $f\colon [0,1]^d \to \mathbb{R}^d$ of radius $\rho < 1/2$ with $\tau' = 4\tau/(1+2\tau)$ or $\tau' = \tau(2+4\rho)$.

Let $G = \{0,1,2,\dots,n\}^d$ be the d-dimensional hypergrid: we identify each point $v \in G$ of the grid with a dataset $x^{(v)} \in (\{0,1\}^d)^n$ in the following fashion. For row $i \in [n]$ and column $j \in [d]$, we have

$$x^{(v)}_{ij} = \mathbb{1}_{\{v_j \ge i\}},$$

where $\mathbb{1}$ denotes the indicator function. That is, if $v = (v_1,\dots,v_d)$, then $x^{(v)}$ is an $n \times d$ matrix where in each column j, the first v_j entries are ones and any remaining entries are zeroes. It follows that $\mathrm{sum}(x^{(v)}) = v$, and if u, v are at ℓ_1 distance at most a, then $x^{(u)}$ and $x^{(v)}$ are at distance at most a from each other as datasets: $D_{\mathrm{Ham}}(x^{(u)}, x^{(v)}) \le \|u - v\|_1$.

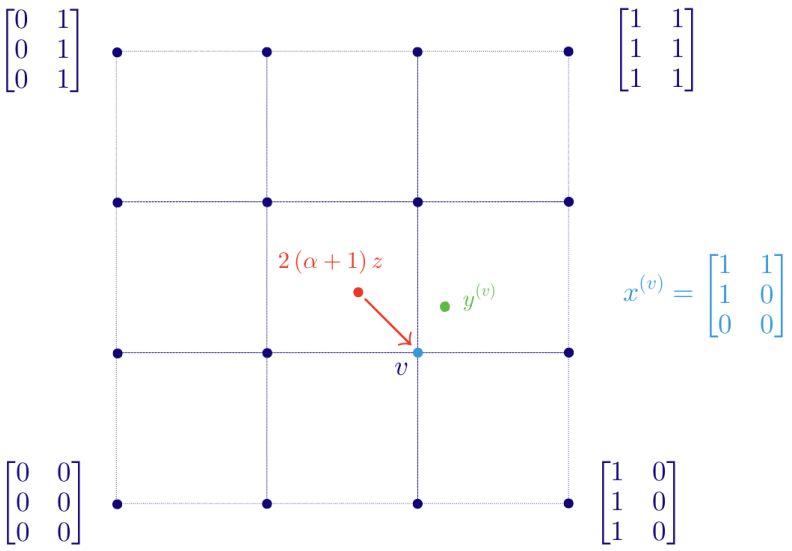


Figure 3 The hypergrid G for $d = 2$ and $n = 3$. Each integer point $v \in G = \{0, 1, 2, 3\}^2$ corresponds to a dataset $x^{(v)}$ (a 3×2 matrix) with the property that $\text{sum}(x^{(v)}) = v$ (the column sums of the matrix at v is just v). The figure shows how a point $z \in [0, 1]^2$, scaled by $2(\alpha + 1)$, is rounded to the nearest v, here $(2, 1)$. The given summation mechanism M takes $x^{(v)}$ to a nearby point $y^{(v)}$. Not shown: this grid is then scaled back down to yield the desired deterministic rounding scheme on $[0, 1]^2$.

Since $H_\infty(M) \le \log_2 K$, for every $v \in G$, there is some output $y^{(v)} \in \mathbb{R}^d$ such that $\Pr[M(x^{(v)}) = y^{(v)}] \ge 1/K$. (If there are several such outputs, choose the lexicographically largest.)

Now we construct our function $f\colon [0, 1]^d \to \mathbb{R}^d$ as follows. For every $z \in [0, 1]^d$, let $v = v(z)$ be a point in G obtained by rounding all coordinates of $2(\alpha + 1) \cdot z$ to the nearest integer (and rounding up if halfway between integers). The resulting v is in G because $2(\alpha + 1) \le n$, and by construction v is at ℓ_∞ distance at most $1/2$ from $2(\alpha + 1) \cdot z$. Then $y^{(v)}$ exists as noted above. See Figure 3. Set

$$f(z) = y^{(v)}/(2\alpha + 2).$$

To show that f is a deterministic rounding scheme, we analyze the two properties separately.

- To show that $\|f(z) - z\|_\infty \le \rho < 1/2$, we use the accuracy of the mechanism. Fix any $z \in [0, 1]^d$: by construction, $2(\alpha + 1) \cdot z$ and $v = v(z)$ are at ℓ_∞ distance at most $1/2$. Recall that we have

  $$\Pr[\|M(x^{(v)}) - \text{sum}(x^{(v)})\|_\infty > \alpha] \le \beta,$$

  and

  $$\Pr[M(x^{(v)}) = y^{(v)}] \ge \frac{1}{K}.$$

  Since $\beta < 1/K$, the outcome of $M(x^{(v)})$ has probability at most $1/K$ of being farther than α from $\text{sum}(x^{(v)})$. Since the particular outcome $y^{(v)}$ has probability greater than $1/K$, then $y^{(v)}$ must be at ℓ_∞ distance at most α from $\text{sum}(x^{(v)}) = v$, and hence at ℓ_∞ distance at most $\alpha + 1/2$ from $2(\alpha + 1) \cdot z$. Hence, $f(z) = y^{(v)}/(2\alpha + 2)$ is at ℓ_∞ distance at most $\rho = \frac{1}{2\alpha+2} \cdot \frac{2\alpha+1}{2} < \frac{1}{2}$ from z.

- For the second, we use the privacy guarantee of the mechanism: consider a closed ℓ_∞ ball $B(w, \tau')$ and points $z_1, \ldots, z_k \in B(w, \tau')$ such that there are k distinct outputs $f(z_1), \ldots, f(z_k)$. Let $v_1, \ldots, v_k$ be the corresponding gridpoints, i.e., $v_i = v(z_i)$. Recall that $f(z_i) = y^{(v_i)}/(2\alpha + 2)$, so the $y^{(v_i)}$'s are all distinct.
  Since $z_1, \ldots, z_k$ are all in $B(w, \tau')$, the gridpoints $v_1, \ldots, v_k$ are all at ℓ_∞ distance at most $2(\alpha+1)\tau' + 1/2$ from $2(\alpha+1)w$, which is itself at ℓ_∞ distance at most $1/2$ from some gridpoint v_0. In particular, each v_i is at ℓ_1 distance at most $d \cdot (2(\alpha+1)\tau' + 1)$ from v_0, so the dataset $x^{(v_i)}$ is at distance at most h from $x^{(v_0)}$ in Hamming distance, for

  $$h = \min(d \cdot (2(\alpha+1)\tau' + 1), n).$$

  By (group) differential privacy, for each $1 \le i \le k$, we have

  $$\Pr[M(x^{(v_0)}) = y^{(v_i)}] \ge e^{-\varepsilon \cdot h} \cdot \Pr[M(x^{(v_i)}) = y^{(v_i)}] - he^{(h-1)\varepsilon}\delta \ge e^{-\varepsilon \cdot h} \cdot \frac{1}{K} - he^{-\varepsilon}\delta.$$

  Summing over $1 \le i \le k$, we get

  $$1 \ge \sum_{i=1}^{k} \Pr[M(x^{(v_0)}) = y^{(v_i)}] \ge e^{-\varepsilon \cdot h} \cdot \frac{k}{K} - khe^{-\varepsilon}\delta.$$

  Reorganizing and upper bounding $e^{-\varepsilon} \le 1$, we have

  $$k \le \frac{K \cdot e^{\varepsilon h}}{1 - hKe^{h\varepsilon}\delta}$$

  as desired.

As mentioned earlier, all that remains is to invoke Lemma 4.3 to lift our deterministic rounding scheme from $[0,1]^d$ to the whole of $\mathbb{R}^d$. ◀

It only remains to establish Lemma 4.3.

Proof of Lemma 4.3. Fix a (k, τ)-deterministic rounding scheme $f\colon [0,1]^d \to \mathbb{R}^d$ of radius $\rho < 1/2$, with $\tau < 1$. Although the image of f might extend to points outside $[0,1]^d$ by ℓ_∞-distance $\rho < 1/2$, we may assume without loss of generality that the image of f is contained in $[0,1]^d$ because projecting the outputs of f to the nearest point in $[0,1]^d$ preserves both properties of a deterministic rounding scheme.

We first modify f so that it behaves nicely near the faces of the cube to get a rounding scheme f', then we show how to extend that to all of $\mathbb{R}^d$ to get the desired rounding scheme f''.

We define $f'\colon [0,1]^d \to [0,1]^d$ by the following process, which is also illustrated in Figure 4.

1. Given $x \in [0,1]^d$, for each $i \in [d]$, set

   $$x'_i = \begin{cases} 0 & \text{if } x_i \le \tau/2 \\ 1 & \text{if } x_i \ge 1 - \tau/2 \\ x_i & \text{otherwise.} \end{cases}$$

   That is, if x is in a "moat" of width $\tau/2$, then it is $\tau/2$-close to one (or possibly several) face(s) of the cube and x' is the projection of x onto (the intersection of) the face(s). Otherwise, set $x' = x$. Note that x' is well-defined by our assumption that $\tau < 1$.

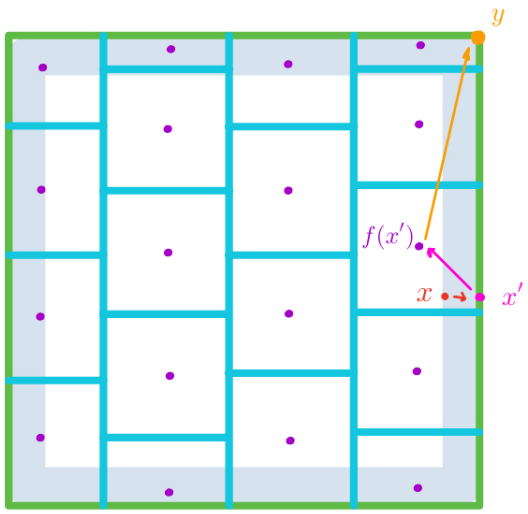


Figure 4 The trajectory of a point in $[0,1]^2$ as it moves from x to x' to $f(x')$ to y. This composition defines the rounding scheme f' on $[0,1]^2$, given any rounding scheme f on $[0,1]^2$. In the case shown, f rounds points in each cell to a point inside. The shaded area is a "moat" of thickness $\tau/2$ where points get projected onto the boundary (and are otherwise not moved) before applying f. The final step is rounding to a corner of $[0,1]^2$.

2. For each $i \in [d]$, set

$$y_i = \begin{cases} 0 & \text{if } 0 \le f(x')_i < \frac{1}{2} \\ 1 & \text{if } \frac{1}{2} \le f(x')_i \le 1. \end{cases}$$

   (Recall that we can assume without loss of generality that the image of f is in $[0,1]^d$.) That is, we round $f(x')$ to the nearest corner of the cube.
3. Set $f'(x) := (y_1, \ldots, y_d) \in \mathbb{R}^d$.

To understand the radius of f', consider Figure 4. The first step from x to x' is bounded (in ℓ_∞ distance) by $\tau/2$ (and is zero if x is not in the moat). The second step from x' to $f(x')$ is bounded by ρ (the radius of f). The third step from $f(x')$ to y is bounded by $1/2$ (since it is rounding to a cube corner). The triangle inequality shows the total distance from x to y is bounded by $1/2 + \rho + \tau/2$, but Figure 4 suggests we can improve this bound: some cancellation is occurring in axes directions where x is close to the boundary. To make this intuition precise, we do casework.

- If $x_i \le \tau/2$ or $x_i \ge 1 - \tau/2$, then since $|f(x')_i - x'_i| \le \rho < 1/2$, we see from y_i's definition that $y_i = x'_i$ so that

$$|y_i - x_i| = |x'_i - x_i| \le \frac{\tau}{2}.$$

- If $\tau/2 < x_i < 1 - \tau/2$, then since $y_i \in \{0,1\}$, we have

$$|y_i - x_i| < 1 - \frac{\tau}{2}.$$

  But when $\tau/2 < x_i < 1 - \tau/2$ we also have $x'_i = x_i$, which implies that

$$|y_i - x_i| \le |(y_i - f(x')_i)| + |f(x')_i - x'_i| \le \frac{1}{2} + \rho.$$

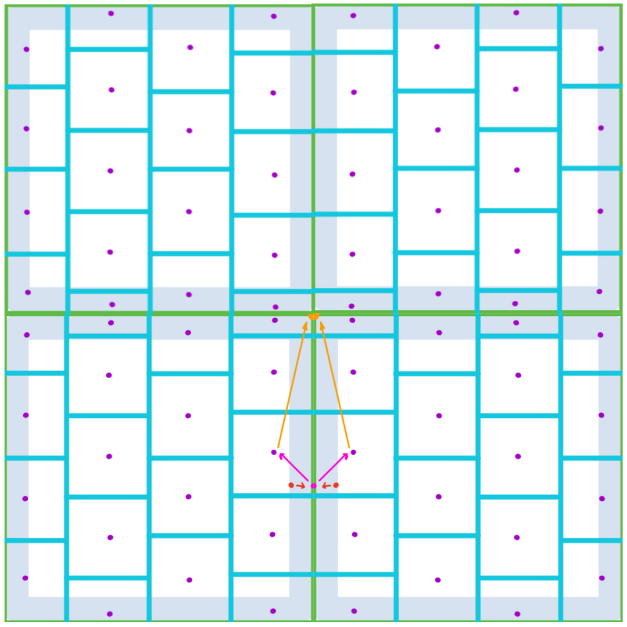


Figure 5 The construction of a rounding scheme f'' on $\mathbb{R}^2$ from a rounding scheme f' on the box $[0,1]^2$ that was illustrated in Figure 4. The definition of f' is reflected across the sides of the box to fill out $\mathbb{R}^2$. Nearby points in the moats of each box have similar fates as their reflections.

Since $\tau < 1$, the bound $\tau/2$ never exceeds either $1 - \frac{\tau}{2}$ or $\frac{1}{2} + \rho$. So, for all x,

$$\|f'(x) - x\|_\infty \leq \min\left\{1 - \frac{\tau}{2}, \frac{1}{2} + \rho\right\},$$

establishing the bound on the radius.

Observe that for every ℓ_∞ ball $B_\infty(z, \tau/2)$ of radius $\tau/2$, we have

$$|f'(B_\infty(z,\tau/2) \cap [0,1]^d)| \leq |f(B_\infty(z,\tau) \cap [0,1]^d)| \leq k.$$

This is because if $\|x - z\|_\infty \leq \tau/2$, then $\|x' - z\|_\infty \leq \tau$, and we obtain $f'(x)$ by applying f to x' and then projecting.

From f', we construct a deterministic rounding scheme $f'' \colon \mathbb{R}^d \to \mathbb{R}^d$ by using f' in each unit cube, but reflected so that points on either side of a unit cube boundary behave similarly. See Figure 5. We define f'' as follows:

Given $x \in \mathbb{R}^d$.

1. For each $i \in [d]$, write $x_i = r_i + s_i e_i$, where $r_i \in 2\mathbb{Z}$, $s_i \in \{\pm 1\}$, and $e_i \in [0,1]$. Note that this is not unique when $x_i \in \mathbb{Z}$, as two decompositions are then possible with either $s_i = 1$ and one with $s_i = -1$ (but both with the same value of $e_i \in \{0,1\}$).[5] We will show that the output $f''(x)$ is independent of the choice we make.
2. Let $e = (e_1, \ldots, e_d) \in [0,1]^d$.
3. For each $i \in [d]$, set $y_i = r_i + s_i f'(e)_i$.
4. Set $f''(x) = (y_1, \ldots, y_d)$.

[5] Specifically: if x_i is an even integer, then (r_i, s_i, e_i) could be either $(x_i, 1, 0)$ or $(x_i, -1, 0)$. If x_i is an odd integer, then the two options are $(x_i - 1, 1, 1)$ and $(x_i + 1, -1, 1)$.

The fact that $f''(x)$ is well-defined, regardless of the non-unique decompositions when x_i is boolean follows from the facts that (a) the vector e is independent of how those choices are made (already noted above), and (b) when $e_i \in \{0,1\}$, we have $e'_i = e_i$ and $f'(e)_i = e'_i$ (shown earlier for points x where x_i close to 0 or 1), so that $y_i = r_i + s_i f'(e)_i = r_i + s_i e_i = x_i$, regardless of whether we chose to use $s_i = 1$ or $s_i = -1$.

We can now analyze the guarantees this f'' provides. For every $x \in \mathbb{R}^d$, since $|s_i| = 1$ for all i we get

$$\|f''(x) - x\|_\infty = \|f'(e) - e\|_\infty \leq \min\left\{\frac{2-\tau}{2}, \frac{1+2\rho}{2}\right\}$$

recalling the radius of f' established earlier. Next, consider any ℓ_∞ ball B of radius at most $\tau/2$. We claim that $|f''(B)| = |f''(B')|$ for an ℓ_∞ ball $B' \subseteq B$ that is entirely contained within a single unit hypercube whose corners are in $\mathbb{Z}^d$, and thus $|f''(B')| \leq k$ by the guarantees of f'. The reason is that if $x \in \mathbb{R}^d$ is within ℓ_∞ distance $\tau/2$ from a face F of a unit hypercube, $f''(x) = f''(x')$, where x' is the projection of x to F. Thus we can remove portions of B that are on the opposite side of a face from its center without changing $f(B)$. (More formally, if we write $B = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_d, b_d]$, then for any dimension i where there is an integer c_i strictly between a_i and b_i, we can replace the interval $[a_i, b_i]$ with the shorter of $[a_i, c_i]$ and $[c_i, b_i]$ without changing $f(B)$.)

Therefore, f'' is a (k, ζ)-deterministic rounding scheme of radius $r = \min\left\{\frac{2-\tau}{2}, \frac{1+2\rho}{2}\right\}$

$$\zeta = \frac{\tau/2}{2r} = \max\left\{\frac{\tau}{2(2-\tau)}, \frac{\tau}{2(1+2\rho)}\right\}. \qquad \blacktriangleleft$$

References

1. Mark Bun, Marco Gaboardi, Max Hopkins, Russell Impagliazzo, Rex Lei, Toniann Pitassi, Satchit Sivakumar, and Jessica Sorrell. Stability is stable: Connections between replicability, privacy, and adaptive generalization. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pages 520–527, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3564246.3585246.
2. Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Differentially private release and learning of threshold functions. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS '15)*, pages 634–649. IEEE, 18–20 october 2015. Full version posted as arXiv:1504.07553. arXiv:1504.07553.
3. Jesus A De Loera, Elisha Peterson, and Francis Edward Su. A polytopal generalization of Sperner's lemma. *Journal of Combinatorial Theory, Series A*, 100(1):1–26, 2002. doi:10.1006/JCTA.2002.3274.
4. Peter Dixon, A. Pavan, Jason Vander Woude, and N. V. Vinodchandran. List and certificate complexities in replicable learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS '23)*, NeurIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.
5. Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography*, volume 3876 of *Lecture Notes in Comput. Sci.*, pages 265–284. Springer, Berlin, 2006. doi:10.1007/11681878_14.
6. Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014. doi:10.1561/0400000042.
7. Simson L. Garfinkel and Philip Leclerc. Randomness concerns when deploying differential privacy. In *WPES@CCS*, pages 73–86. ACM, 2020. doi:10.1145/3411497.3420211.

8 Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. User-level differentially private learning via correlated sampling. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 20172–20184, 2021. URL: `https://proceedings.neurips.cc/paper/2021/hash/a89cf525e1d9f04d16ce31165e139a4b-Abstract.html`.

9 Ilya Mironov, Omkant Pandey, Omer Reingold, and Salil Vadhan. Computational differential privacy. In S. Halevi, editor, *Advances in Cryptology—CRYPTO '09*, volume 5677 of *Lecture Notes in Computer Science*, pages 126–142. Springer-Verlag, 16–20 august 2009. `doi:10.1007/978-3-642-03356-8_8`.

10 Salil P. Vadhan. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*, pages 347–450. Springer International Publishing, 2017. `doi:10.1007/978-3-319-57048-8_7`.

11 Jason Vander Woude, Peter Dixon, Aduri Pavan, Jamie Radcliffe, and N. V. Vinodchandran. Geometry of rounding. *CoRR*, abs/2211.02694, 2022. `doi:10.48550/arXiv.2211.02694`.

12 Jason Vander Woude, Peter Dixon, Aduri Pavan, Jamie Radcliffe, and N. V. Vinodchandran. Geometry of rounding: Near optimal bounds and a new neighborhood Sperner's lemma. *CoRR*, abs/2304.04837, 2023. `doi:10.48550/arXiv.2304.04837`.

Coresets for 1-Center in ℓ_1 Metrics

Amir Carmel
Weizmann Institute of Science, Rehovot, Israel

Chengzhi Guo
Peking University, China

Shaofeng H.-C. Jiang
Peking University, China

Robert Krauthgamer
Weizmann Institute of Science, Rehovot, Israel

Abstract

We explore the applicability of coresets – a small subset of the input dataset that approximates a predefined set of queries – to the 1-center problem in ℓ_1 spaces. This approach could potentially extend to solving the 1-center problem in related metric spaces, and has implications for streaming and dynamic algorithms.

We show that in ℓ_1, unlike in Euclidean space, even weak coresets exhibit exponential dependency on the underlying dimension. Moreover, while inputs with a unique optimal center admit better bounds, they are not dimension independent. We then relax the guarantee of the coreset further, to merely approximate the value (optimal cost of 1-center), and obtain a dimension-independent coreset for every desired accuracy $\epsilon > 0$. Finally, we discuss the broader implications of our findings to related metric spaces, and show explicit implications to Jaccard and Kendall's tau distances.

2012 ACM Subject Classification Theory of computation → Sketching and sampling; Theory of computation → Streaming models

Keywords and phrases clustering, k-center, minimum enclosing balls, coresets, ℓ_1 norm, Kendall's tau, Jaccard metric

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.28

Funding *Shaofeng H.-C. Jiang*: Research partially supported by a national key R&D program of China No. 2021YFA1000900 and a startup fund from Peking University.
Robert Krauthgamer: Work partially supported by the Israel Science Foundation grant #1336/23, by the Israeli Council for Higher Education (CHE) via the Weizmann Data Science Research Center, and by a research grant from the Estate of Harry Schutzman. Part of this work was done while visiting the Simons Institute for the Theory of Computing.

1 Introduction

Clustering is a fundamental task in unsupervised learning and data analysis in general, with wide-ranging applications. Typically, the input is a dataset of n points in $\mathbb{R}^d$ or some other feature space of dimension d, and the objective is to partition the dataset into clusters, each characterized by a high degree of similarity. In the current era of big data, both the number of points and their dimension are often excessive, making the computational demands significant. In center-based clustering, the goal is to further assign to each cluster a representative "center" point. The number of centers (and ergo clusters) is often specified in advance, and denoted by k, giving rise to the fundamental k-center, k-median and k-mean problems, which are famous for their simplicity and broad applicability. The case $k = 1$ is particularly important as the most basic setting, and focuses not on partitioning the points but rather on aggregating them, by identifying a representative that best captures the entire dataset.


© Amir Carmel, Chengzhi Guo, Shaofeng H.-C. Jiang, and Robert Krauthgamer;
licensed under Creative Commons License CC-BY 4.0



We study the 1-center problem in ℓ_1 metrics (i.e., Manhattan distance). The motivation for ℓ_1 metrics is twofold. First, ℓ_1 metrics arise in many practical settings, especially when the dimension is large. Second, common discrete metric spaces, such as the Hamming, Kendall's tau, and Jaccard metrics, are closely related to ℓ_1, as they can be embedded into $\mathbb{R}^d$ endowed with the ℓ_1 metric with all distances preserved isometrically (i.e., with no distortion).

The 1-center problem has been investigated extensively in many metric spaces. In the context of strings, 1-center under the Hamming distance and Edit Distance is a classical problem known as the Closest String Problem [26, 27, 29]. In the context of permutations (a common model for rankings), 1-center under Kendall's tau distance is known as the maximum rank aggregation problem [5, 32]. The 1-center problem has also been studied in other discrete metrics, such as the Ulam distance between permutations and the Jaccard distance between sets [9, 13]. Additionally, recent work studied the 1-center problem under various metrics, including ℓ_p, edit distance, and Ulam, specifically to examine how the time complexity depends on the dimension d [1]. In fact, the 1-center problem is known to be NP-hard in many metric spaces, and it seems that each one requires distinct algorithmic techniques. For some metrics, a PTAS is known, while for others, the existence of a PTAS or even a non-trivial approximation algorithm remains an open question and an active area of research.

A coreset is a data-summarization concept, where the input dataset is replaced by a small subset that approximates its clustering properties. It can lead to significant reductions in computational resources, particularly storage and communication, and plays a pivotal role in modern algorithmic settings, such as streaming, distributed, and dynamic. It is thus crucial to understand the tradeoff between the size of a coreset and its accuracy (approximation factor). Coresets have been studied intensely since their introduction by [2], see the surveys [3, 21, 28, 31], and they have been successfully applied in a broad range of settings, for example different clustering objectives and metric spaces. Surprisingly, little is known about coresets in ℓ_1, particularly in high dimension, leaving significant gaps in our understanding. Small coresets in ℓ_1, if they exist and can be computed efficiently, could potentially pave the way for a unified approach for 1-center also in related metrics, like Kendall's tau and Ulam. Moreover, studying coresets for these metrics may provide insights into the geometric structure of these metric spaces.

1.1 Problem setup

In the 1-*center problem*, also known as *Minimum Enclosing Ball (MEB)*, the input is a set P of points in a metric space (X, dist), and the goal is to find a center point $c \in X$ that minimizes the objective function $\mathrm{cost}(c, P) := \max_{p \in P} \mathrm{dist}(p, c)$. Denote the optimal value by $\mathrm{opt}(P) := \min_{c \in X} \mathrm{cost}(c, P)$, and observe that it is monotone, i.e., $Q \subseteq P$ implies $\mathrm{opt}(Q) \leqslant \mathrm{opt}(P)$. While we mostly consider the space $(\mathbb{R}^d, \ell_1)$, these definitions capture also other metrics, like Kendall's tau distance where X is the set of permutations over $[d]$.

There are different (and sometimes inconsistent) definitions for coresets, which were usually developed when the desirable guarantees were deemed impossible or difficult to prove, leading to more relaxed guarantees. We face this same issue (especially for high dimension), and thus consider a sequence of definitions. For brevity, we present only the definitions most relevant to our work, and do not discuss possible generalizations (e.g., to k-center) or variants that appear in the literature. A common and very useful type of coreset is a *strong coreset*, which approximates the cost of every center, as follows.

▶ **Definition 1.1** (Strong Coreset). *A subset $Q \subseteq P$ is a* strong ϵ-coreset *for a 1-center instance $P \subseteq X$ if*

$$\forall c \in X, \qquad \mathrm{cost}(c, P) \leqslant (1+\epsilon)\,\mathrm{cost}(c, Q). \tag{1}$$

The guarantee in (1) is extremely effective in reducing (the task of solving) P to Q, with a small loss in the objective value. In Euclidean spaces, strong coresets for 1-center are known, however their size is exponential in the dimension, namely, $|Q| \leqslant (1/\epsilon)^{O(d)}$ [3], and unfortunately this upper bound is existentially tight.[1] The upper bound actually extends to every metric space with doubling dimension d [4, 10, 20], and therefore applies to $\mathbb{R}^d$ endowed with the ℓ_1 norm (or every other norm).

It is thus natural to ask: Can the coreset size be improved specifically for ℓ_1 norm? If not, can weaker notions lead to polynomial (in d) or even dimension-independent size?

1.2 Our results

Weak coresets. We point our attention to a more relaxed notion, called *weak coreset*, which only approximates the cost of certain centers, namely, centers that may be found by solving the coreset instance. Trivially, every strong ϵ-coreset is also a weak ϵ-coreset.

▶ **Definition 1.2** (Weak Coreset). *A subset $Q \subseteq P$ is a* weak ϵ-coreset *for a 1-center instance $P \subseteq X$ if*

$$\forall c^Q \in \operatorname*{argmin}_{c \in X} \mathrm{cost}(c, Q), \qquad \mathrm{cost}(c^Q, P) \leqslant (1+\epsilon)\,\mathrm{cost}(c^Q, Q). \tag{2}$$

Weak coresets were introduced in [8] specifically for 1-center in Euclidean space. It is well-known that in Euclidean space, there is a unique optimal solution,[2] which is why previous work often refers to c^Q as *the* optimal center. The challenge then becomes how to extend this definition to other metric spaces, like ℓ_1 norm, that might have multiple optimal solutions. The typical use of a coreset Q is to apply an algorithm on Q and take the resulting center c^Q as an approximated solution for the original instance P. Since a generic algorithm might pick any of the multiple optimal centers for Q, it is essential to have a guarantee that applies to every possible optimal solution c^Q for Q, hence the universal quantifier ("for all") in (2).

Remarkably, in Euclidean space, every instance P admits a weak ϵ-coreset of size $\lceil 1/\epsilon \rceil$, which is independent of the input size $n = |P|$ and of the Euclidean dimension d, and is in fact tight [7]. This result improved an earlier $O(1/\epsilon^2)$ bound from [8], which was also dimension independent. Can similar bounds be proved for ℓ_1 metrics?

We demonstrate that, in sharp contrast to the ℓ_2 setting, weak coresets in ℓ_1 are dimension dependent and moreover grow exponentially with d, even for a fixed $\epsilon > 0$. At the same time, we show how to construct a strong 0-coreset of size 2^d, which again contrasts with the ℓ_2 setting (and doubling metric spaces), where coresets grow with $1/\epsilon$ and are thus not applicable for $\epsilon = 0$. It is well-known that a strong coreset is *composable*, i.e., the union of coresets is itself a coreset for the union of the original datasets, which is very useful for designing algorithms in the streaming and dynamic settings. Thus, our coreset construction can be used in these settings, particularly in low dimension.

[1] There is a folklore lower bound, which follows by considering an instance P formed by an ϵ-net of the unit sphere S^{d-1}; every strong coreset must contain all of P and thus have size $(1/\epsilon)^{\Omega(d)}$.

[2] The uniqueness follows from the fact that ℓ_2 is strictly convex, whereas ℓ_1 is convex but not strictly convex.

▶ **Theorem 1.3.** *Consider the* 1*-center problem in* $\mathbb{R}^d$ *under* ℓ_1 *norm.*
(a) *Every instance* $P \subset \mathbb{R}^d$ *admits a strong* 0*-coreset of size* 2^d*.*
(b) *There exists* $P \subset \mathbb{R}^d$*, such that every weak* ϵ*-coreset for* $\epsilon < \frac{1}{3}$ *must have size* $2^{\Omega(d)}$*.*

We prove this theorem in Section 2. The upper bound leverages the unique geometry of ℓ_1, where only 2^d directions are in effect important. The lower bound builds a hard instance and employs classical techniques from coding theory, particularly the Hamming bound. We remark that an unpublished result from [30, Theorem 5], for a related problem about a convex shape in ℓ_2 metric, seems to imply a weak ϵ-coreset of size $\text{poly}(d/\epsilon)$ for our problem of 1-center in ℓ_1 metrics. This would contradict our lower bound in Theorem 1.3, indicating that its statement is inaccurate or its proof is flawed; see also a similar discussion in [6].

Due to the exponential dependency on d in the general case, it is important to identify cases with a smaller coreset size, say $\text{poly}(d)$ or even dimension independent. Additionally, one may suspect that the huge disparity between ℓ_1 and Euclidean space, which does admit a dimension-independent weak coreset [7, 8], is the fact that in ℓ_1 the optimal center need not be unique. We thus study the special case of ℓ_1 where the optimal center is unique, which may occur naturally in certain contexts, or be "enforced" by adding at most two points to the input dataset. We provide bounds that are exponentially smaller than in the general case, but still depend on d, which establishes that ℓ_1 is inherently more complex than ℓ_2.

▶ **Theorem 1.4.** *Consider the* 1*-center problem in* $\mathbb{R}^d$ *under* ℓ_1 *norm, on instances that have a unique optimal center.*
(a) *Every instance* $P \subset \mathbb{R}^d$ *with unique optimum admits a weak* 0*-coreset of size* $2d$*.*
(b) *There exists* $P \subset \mathbb{R}^d$ *with unique optimum, such that every weak* ϵ*-coreset for fixed* $\epsilon < 1$ *must have size* $\Omega(\log d)$*.*

We prove this theorem in Section 3. We further conjecture that the lower bound can be improved to $\Omega(d)$, and prove it in the special case $\epsilon = 0$ using Hadamard code. The upper bound in our theorem employs tools from convex geometry, specifically the Steinitz Theorem, which is relatively obscure, to prove the existence non-constructively (i.e., without an efficient algorithm). For the lower bound, we consider the instance $P = \{\pm 1\}^d$, which has a unique optimal center at the origin, and show that a coreset Q that is too small, must have an optimal center c^Q in which most coordinates are ± 1, and this center is clearly a poor solution for P.

Value-preserving coreset. To further reduce the coreset size, we consider an even less restrictive variant that preserves solely the objective function $\text{opt}(P)$, without imposing any conditions on the optimal centers of the coreset. This concept, which we shall refer to as a *value-preserving coreset*, has been studied previously under the general term "coreset" [6, 14, 33], and it is most useful in applications that focus on measuring the similarity among points in the cluster, rather than identifying a center. It is easy to verify that every weak coreset is also a value-preserving coreset.

▶ **Definition 1.5** (Value-Preserving Coreset)**.** *A subset* $Q \subseteq P$ *is a* value-preserving ϵ-coreset *for a* 1*-center instance* $P \subseteq X$ *if*

$$\text{opt}(P) \leqslant (1+\epsilon)\,\text{opt}(Q). \tag{3}$$

We show that there exist value-preserving ϵ-coresets of size $\tilde{O}(1/\epsilon^2)$, which is dimension independent, in contrast with Theorems 1.3 and 1.4. We thus establish a sharp separation between preserving an optimal solution (center point) or only preserving its value. This result

is stated in the next theorem and proved in Section 4. It also answers a question posed in [6], which studies a generalization of 1-center, called the Minimum Enclosing Polytope problem, and asks for polytopes that admit a dimension-independent value-preserving coreset.[3]

▶ **Theorem 1.6.** *Consider the* 1*-center problem in* $\mathbb{R}^d$ *under* ℓ_1 *norm. Every instance* $P \subset \mathbb{R}^d$ *admits a value-preserving* ϵ*-coreset of size* $O(\frac{1}{\epsilon^2} \log \frac{1}{\epsilon})$.

We prove this theorem in Section 4. Our algorithm is based on random sampling, a technique that was rarely used (if at all) for 1-center coresets, but widely used for k-median and k-means coresets [15, 18, 22]. This is very natural because sampling is an excellent fit for objectives like k-median that that are formed by summation, but not for objectives like k-center that are formed by maximization, which are sensitive to missing even one term. It is thus not surprising that our sampling is conducted not on the 1-center problem but rather on its dual. More precisely, we formulate 1-center as a linear programming (LP) problem and write its dual problem, then solve this dual problem to obtain sampling probabilities for the input points. Crucially, the dual objective is formed by summation, and is thus conducive to sampling. Compared with the primal-dual framework, our algorithm is dual-only and it solves the dual LP explicitly, which is then related in the analysis to the primal via strong duality. Our sampling is similar to the randomized rounding commonly used in approximation algorithms, however its purpose here is to generate a coreset instead of a solution, and also the analysis is somewhat different.

Related metrics and applications. We demonstrate that our results for weak coresets in ℓ_1 (in Sections 2 and 3) apply also to other discrete metric spaces. Specifically, we provide in Section 5 examples that use Kendall's tau and Jaccard metrics, drawing inspiration from the results obtained in the ℓ_1 setting.

1.3 Related work

Another closely related aggregation problem is 1-median. This problem, unlike 1-center, can be solved in ℓ_1 metrics easily, by simply taking in each coordinate the median value of the input points. Utilizing this structure, 1-median in ℓ_1 metrics has a weak ϵ-coreset of size $O(\epsilon^{-2} \log(1/\epsilon))$, which is independent of the dimension [19], in sharp contrast to 1-center. The 1-median problem also admits a PTAS (approximation arbitrarily close to 1 in polynomial time) in many other metrics, including Kendall's tau distance [25], Jaccard distance [16], and a near-linear time PTAS in ℓ_2 [17]. However, it is unclear if a PTAS exists for edit distance, or even for a special case called Ulam distance, and the best upper bound is $(2 - \rho)$-approximation for some fixed $\rho > 0$ [11, 12].

2 Weak ϵ-coresets in ℓ_1

2.1 Construction of strong 0-coreset of size 2^d

In this section we will prove the following theorem:

▶ **Theorem 2.1.** *Every instance* $P \subset \mathbb{R}^d$ *admits a strong* 0*-coreset of size* 2^d.

[3] Prior to our result showing that the ℓ_1 ball (also known as the cross-polytope) admits dimension-independent value-preserving coreset, the only known example was the parallelotope, which trivially has a value-preserving 0-coreset with only 2 points.

Proof. Let $\Sigma = \{-1, 1\}^d$ denote the set of all sign vectors in $\mathbb{R}^d$. For each sign vector $\sigma \in \Sigma$, let p_σ be a point in P such that the inner product $p_\sigma \cdot \sigma$ is maximized. Define $Q_\Sigma = \{p_\sigma : \sigma \in \Sigma\}$, clearly $|Q_\Sigma| \leqslant |\Sigma| = 2^d$.

Let $c \in \mathbb{R}^d$. Since Q_Σ is a subset of P, we have that $\text{cost}(c, Q_\Sigma) \leqslant \text{cost}(c, P)$. Let p be some point in P, then there exists a sign vector $\sigma \in \Sigma$ for which we can write the distance between p and c as:

$$\|p - c\|_1 = \sigma \cdot (p - c) = \sigma \cdot p - \sigma \cdot c$$

By definition of Q_Σ, there exists a point $p_\sigma \in Q_\Sigma$ satisfying $\sigma \cdot p \leqslant \sigma \cdot p_\sigma$. Thus,

$$\|p - c\|_1 = \sigma \cdot p - \sigma \cdot c \leqslant \sigma \cdot p_\sigma - \sigma \cdot c \leqslant \sigma \cdot (p_\sigma - c) \leqslant |\sigma \cdot (p_\sigma - c)| \leqslant \|p_\sigma - c\|_1 \leqslant \text{cost}(c, Q_\Sigma).$$

That is,

$$\text{cost}(c, P) = \max_{p \in P} \|p - c\|_1 \leqslant \text{cost}(c, Q_\Sigma)$$ ◀

2.2 Lower bound of $2^{\Omega(d)}$ for weak ϵ-coresets

In this section, we present a set $P \subseteq \mathbb{R}^d$ of size $2^{d(1-o(1))}$ that has only a trivial 0-coreset, namely, only itself. Furthermore, weak ϵ-coresets of P, for $\epsilon \in (0, \frac{1}{3})$, must have size $2^{\Omega(d)}$.

▶ **Theorem 2.2.** *There exists a set $P \subseteq \mathbb{R}^d$ of size $2^{d(1-o(1))}$, such that*

- *if $Q \subseteq P$ is a weak 0-coreset of P, then $Q = P$; and*
- *for all $\epsilon \in [0, \frac{1}{3})$, if $Q \subseteq P$ is a weak ϵ-coreset of P, then $|Q| \geqslant 2^{\Omega(d)}$.*

Proof. We may assume without loss of generality that d is even, as otherwise we can take the construction for $d - 1$ and append 0 to all the points (vectors).

Denote by $\vec{1}$ the vector $(1, 1, \ldots, 1)$, and let $B \subseteq \{\pm 1\}^d$ be the set of points that are balanced, in the sense that they have an equal number of 1 and -1 coordinates, i.e., $B = \{b \in \{\pm 1\}^d : \sum_{i=1}^d b_i = 0\}$. Fix $0 < \delta \leqslant \frac{1}{3}$, and let $P = \{-\vec{1}, \vec{1}\} \bigcup (1-\delta)B$, where $(1-\delta)B$ denotes the set of points in B scaled by factor $1 - \delta$. Thus, $|P| = 2 + \binom{d}{d/2} = \Theta(2^d/\sqrt{d})$. One can verify that the origin $\vec{0} \in \mathbb{R}^d$ is an optimal center, with value d. However it is not unique; to see this, observe that the antipodal pair $\{-\vec{1}, \vec{1}\}$ establishes the optimal value $\text{opt}(P) = d$, thus an optimal center must have the same distance d to each of them, which holds for points $x \in [-1, 1]^d$ that lie on the hyperplane $\sum_{i=1}^d x_i = 0$. The set $(1 - \delta)B$ restricts the optimal solutions (inside that hyperplane) in a delicate manner, as we shall see.

We first show that P is the only weak 0-coreset of P. Let $Q \subseteq P$ be a weak 0-coreset, and notice that $\text{opt}(Q) = \text{opt}(P) = d$, implying that Q must contain the antipodal pair $-\vec{1}$ and $\vec{1}$. Assume towards contradiction that Q is a proper subset of P. Then there exists $(1 - \delta)\tilde{b} \in P \backslash Q$ for some $\tilde{b} \in B$. Let $\eta = \frac{1}{d}$ and consider $c^* = -(1 + \eta)\delta\tilde{b}$; it cannot be an optimal center of P, because

$$\left\|-(1 + \eta)\delta\tilde{b} - (1 - \delta)\tilde{b}\right\|_1 = \left\|(-1 - \eta\delta)\tilde{b}\right\|_1 = (1 + \eta\delta)\left\|\tilde{b}\right\|_1 > d.$$

We next show that this point $c^* = -(1+\eta)\delta\tilde{b}$ is an optimal center of Q. Indeed, observe that $\delta \leqslant \frac{1}{3}$ implies $(1 + \eta)\delta \leqslant 1 - \delta$, and thus the sign of $1 - \delta$ will not change if we add/subtract to it $(1 + \eta)\delta$. Let $\text{sgn}(x) \in \{-1, 0, 1\}$ denote the sign function. Now consider $a \in Q$; its distance to $-(1 + \eta)\delta\tilde{b}$ is:

$$\left\|a+(1+\eta)\delta\tilde{b}\right\|_1 = \sum_{i=1}^{d}\left|a_i+(1+\eta)\delta\tilde{b}_i\right| = \sum_{i=1}^{d}\operatorname{sgn}(a_i)\left(a_i+(1+\eta)\delta\tilde{b}_i\right)$$
$$= \|a\|_1 + (1+\eta)\delta\sum_{i=1}^{d}\operatorname{sgn}(a_i)\tilde{b}_i. \tag{4}$$

In the case $a \in \{-\vec{1},\vec{1}\}$, the above equals d. In the remaining case $a \in Q\backslash\{-\vec{1},\vec{1}\}$, the choice of $\tilde{b}$ implies that $\sum_{i=1}^{d}\operatorname{sgn}(a_i)\tilde{b}_i \leqslant d-2$, and thus

$$\left\|a+(1+\eta)\delta\tilde{b}\right\|_1 \leqslant \|a\|_1 + (1+\eta)\delta(d-2) \leqslant (1-\delta)d + (1+\eta)\delta(d-2) < d.$$

We have thus confirmed that $-(1+\eta)\delta\tilde{b}$ is an optimal center of Q, but not for P. This contradicts our assumption that Q is a weak 0-coreset, and completes the proof of the first item.

To prove the second item, let $P \subset \mathbb{R}^d$ be as before but now for $\delta = \frac{1}{3}$, and consider a weak ϵ-coreset Q of P, for some $0 < \epsilon < \delta = \frac{1}{3}$. Notice that Q must contain both $-\vec{1}$ and $\vec{1}$, as otherwise, if $-\vec{1},\vec{1} \notin Q$, consider the origin $\vec{0} \in \mathbb{R}^d$ as a center. If Q contains just one of $\{-\vec{1},\vec{1}\}$ say $\vec{1}$, consider $\delta\vec{1}$ as a center. In both cases, we obtain that $\operatorname{opt}(Q) \leqslant (1-\delta)d < (1-\epsilon)d < \frac{1}{1+\epsilon}\operatorname{opt}(P)$, in contradiction to the weak ϵ-coreset.

Denote by $\psi(Q)$ the minimum Hamming distance between any point in B and its furthest point in Q, that is, $\psi(Q) = \min_{b\in B}\max_{a\in Q}\operatorname{Hamm}(a,b)$. The next proposition follows by a standard counting-bound.

▶ **Proposition 2.3.** *If* $|Q| \leqslant \frac{1}{\sqrt{2d}}2^{0.18d}$, *then* $\psi(Q) < \frac{3}{4}d$.

Proof. Denote by $C_{\frac{3d}{4}}(a)$ the set of all points in $\{\pm1\}^d$ whose Hamming distance from $a \in \{\pm1\}^d$ is at least $\frac{3d}{4}$. Then

$$\left|C_{\frac{3d}{4}}(a)\right| \leqslant \sum_{0\leqslant j\leqslant\frac{d}{4}}\binom{d}{j} \leqslant 2^{dH(\frac{1}{4})},$$

where the last inequality is the known entropy bound for binomial coefficients and $H(p) = -p\log p - (1-p)\log(1-p)$, hence $H(\frac{1}{4}) \approx 0.811$. Taking a union over all $a \in Q$,

$$\left|\bigcup_{a\in Q}C_{\frac{3d}{4}}(a)\right| \leqslant \sum_{a\in Q}\left|C_{\frac{3d}{4}}(a)\right| \leqslant 2^{dH(\frac{1}{4})}|Q| < \binom{d}{d/2} = |B|,$$

where in the last inequality we used the known bound $\frac{1}{\sqrt{2d}}2^d \leqslant \binom{d}{d/2}$. Hence, there exists some point in B for which the maximum Hamming distance to points in Q is smaller than $\frac{3d}{4}$. ◀

Assume towards contradiction that $|Q| \leqslant \frac{1}{\sqrt{2d}}2^{0.18d}$. Then by Proposition 2.3, there is a point $\tilde{b} \in B$ for which the maximum Hamming distance to points in Q is smaller than $\frac{3d}{4}$. We next show that the point $2\delta\tilde{b}$ is an optimal center for Q. Similarly to Equation (4), since $2\delta \leqslant 1-\delta$, for every $a \in Q$ we can write:

$$\left\|a-2\delta\tilde{b}\right\|_1 = \sum_{i=1}^{d}\left|a_i-2\delta\tilde{b}_i\right| = \|a\|_1 - 2\delta\sum_{i=1}^{d}\operatorname{sgn}(a_i)\tilde{b}_i,$$

where:

$$\sum_{i=1}^{d} \operatorname{sgn}(a_i)\tilde{b}_i = \sum_{i:\operatorname{sgn}(a_i)=\operatorname{sgn}(\tilde{b}_i)} 1 - \sum_{i:\operatorname{sgn}(a_i)\neq\operatorname{sgn}(\tilde{b}_i)} 1 = d - 2\operatorname{Hamm}(a,\tilde{b}),$$

and thus:

$$\left\|a - 2\delta\tilde{b}\right\|_1 = \|a\|_1 - 2\delta d + 4\delta \operatorname{Hamm}(a,\tilde{b}).$$

In the case $a \in \{-\vec{1},\vec{1}\}$, since $\tilde{b} \in B$ then we have $\left\|a - 2\delta\tilde{b}\right\|_1 = \|a\|_1 = d$. Otherwise $a \in (1-\delta)B$, and $\left\|a - 2\delta\tilde{b}\right\|_1 < \|a\|_1 - 2\delta d + 4\delta \cdot \frac{3}{4}d \leqslant (1-\delta)d + \delta d = d$. In both cases, the distance from $a \in Q$ to $2\delta\tilde{b}$ is at most $d = \operatorname{opt}(Q)$, with equality for some $a \in Q$. Thus, $2\delta\tilde{b}$ is an optimal center for Q. However, $\operatorname{cost}(2\delta\tilde{b}, P) \geqslant \left\|2\delta\tilde{b} + (1-\delta)\tilde{b}\right\|_1 = (1+\delta)d > (1+\epsilon)\operatorname{opt}(P)$. This contradicts that Q is a weak ϵ-coreset of P, and completes the proof of Theorem 2.2. ◀

3 Weak ϵ-coresets in ℓ_1 for inputs with a unique solution

In this section we consider the special case, where the input P has a unique optimal center. We show that even in this special case, the size of the coreset depends on the dimension and provide a lower bound of $\Omega(\log d)$ on the size of every ϵ-coreset for any fixed $\epsilon < 1$. This restriction over ϵ is necessary, as for larger values of ϵ, a naive coreset construction (based on Gonzalez algorithm for k-center [23]) with only 2 points achieves a 3-approximation.[4] We also prove that there always exists a weak 0-coreset of size at most $2d$. This is in stark contrast to Section 2, where we show that in the general case of multiple solutions, 0-coresets might require size exponential in d. Lastly, we present a set such that every weak 0-coreset of this set is of size $2d$, hence the upper bound of $2d$ is tight for the case $\epsilon = 0$.

3.1 Lower bound of $\Omega(\log d)$ for weak ϵ-coresets

▶ **Theorem 3.1.** *Fix $\epsilon < 1$ and consider the set of points $P = \{\pm 1\}^d \subset \mathbb{R}^d$. Then P has a unique optimal center and every weak ϵ-coreset of P must have $\Omega(\log d)$ points.*

Proof. It is easy to note that the origin $\vec{0} \in \mathbb{R}^d$ is a unique optimal center with value d as the input contains a point from every face of the ℓ_1 ball centered at $\vec{0}$. Consider a weak ϵ-coreset $Q \subset P$ of size $|Q| \leqslant \frac{1}{2}\log d$. Every point $p \in Q$ induces a partition Π_p of $[d]$ into at most two parts, by splitting the coordinates of p into the positive and negative ones. Let Π be the common refinement of all these partitions (i.e. placing $i, j \in [d]$ in the same part of Π if and only if for every $p \in P$ they are in the same part of Π_p). Since each Π_p has at most two parts, Π has at most $2^{|Q|} \leqslant \sqrt{d}$ parts. Observe that for every point $p \in Q$, coordinates in the same part of Π have the same sign.

Consider an optimal center c^* for Q. The idea is to modify it iteratively, without increasing its cost for Q, so as to have more coordinates take values in $\{-1, 1\}$. Each iteration takes two indices i, j in the same part of Π, whose coordinates in c^* are *both* not in $\{-1, 1\}$, say, $-1 < c^*_i \leqslant c^*_j < 1$. The iterations stop when no such indices exist. Now add δ to c^*_i and $-\delta$ to c^*_j, where $\delta > 0$ is as large as possible while keeping both new values in the range

[4] Pick arbitrary $p \in P$, then a point $p' \in P$ that is furthest from p, and take $Q = \{p, p'\}$ as the coreset. Note that $\operatorname{opt}(Q) = \frac{1}{2}\left\|p - p'\right\|_1$, and $\frac{1}{2}(p + p')$ is a possible optimal center. For every optimal center c^* for Q, we can write $\operatorname{cost}(c^*, P) \leqslant \left\|c^* - p\right\|_1 + \operatorname{cost}(p, P) \leqslant 1.5\left\|p - p'\right\|_1 = 3\operatorname{opt}(Q) \leqslant 3\operatorname{opt}(P)$.

$[-1,1]$, that is, $\delta = \min\{1 - c_j^*, c_i^* - 1\}$. It is easy to see that after this modification, at least one of c_i^* and c_j^* will be in $\{\pm 1\}$, and at the same time the distance to every $p \in Q$ does not change. The iteration stop at a center $\bar{c}^*$, where in each part of Π at most one coordinate is not in $\{\pm 1\}$, and in total at most $|\Pi| \leqslant 2^{|Q|} \leqslant \sqrt{d}$ coordinates are not in $\{\pm 1\}$.

Finally, we show that $\bar{c}^*$, which is an optimal center for Q, has a too large cost for P. Consider the point in P that is closest to $-\bar{c}^*$, i.e., where each coordinate of $-\bar{c}^*$ is rounded to the nearest value among $\{\pm 1\}$. This point disagrees with $-\bar{c}^*$ (i.e., rounding was "needed") on at most $|\Pi|$ coordinates, hence its distance from $\bar{c}^*$ is at least $2d - |\Pi|$, hence $\operatorname{cost}(\bar{c}^*, P) \geqslant 2d - |\Pi| \geqslant 2d - \sqrt{d} > (1+\epsilon)\operatorname{opt}(P)$. ◀

We remark that the proof works also for the discrete space $\{-1, 0, +1\}^d$ (endowed with ℓ_1 norm), which has more limited choices for the center point, e.g., for c^* above. Moreover, note that the distance of the modified optimal center $\bar{c}^*$ from $\vec{0}$ is at least $d - 2^{|Q|}$.

3.2 Upper bound of $2d$ for weak 0-coresets

We will show that if P has a unique solution, then P has a weak 0-coreset of size at most $2d$.

Two standard notations from the domain of convex geometry are the convex-hull and the interior. The convex hull of P, denoted $\operatorname{conv}(P)$, consists of all convex combinations of points in P, that is every point in $\operatorname{conv}(P)$ can be expressed as a weighted sum of points from P with non-negative weights summing to 1. The interior of P, denoted $\operatorname{int}(P)$, refers to the set of points that lie strictly inside the convex hull. In other words, it consists of points that can be surrounded by a small ball completely contained within $\operatorname{conv}(P)$.

An important tool we use is Steinitz's theorem [24].

▶ **Theorem 3.2** (Steinitz's theorem). *Let $\vec{0} \in \operatorname{int}(\operatorname{conv}(S))$ for some $S \subseteq \mathbb{R}^d$. Then there exists $R \subseteq S$ of size at most $2d$, such that $\vec{0} \in \operatorname{int}(\operatorname{conv}(Q))$.*

Next we introduce the notion of complete set, we will then provide an alternative formulation of Steinitz's theorem that will be useful in our application.

▶ **Definition 3.3.** *A set $S \subseteq \mathbb{R}^d$ is complete if for every $v \in \mathbb{R}^d \backslash \{\vec{0}\}$ there exists $s \in S$ with $v \cdot s > 0$.*

▶ **Lemma 3.4.** *$\vec{0} \in \operatorname{int}(\operatorname{conv}(S))$ if and only if S is complete.*

Proof. ($\Rightarrow$) : Write $\vec{0}$ as a convex combination, $\vec{0} = \sum_i \lambda_i s_i$ where $\lambda_i \geqslant 0$ and $\sum_i \lambda_i = 1$. Then,

$$0 = v \cdot \vec{0} = v \cdot (\sum_i \lambda_i s_i) = \sum_i \lambda_i (v \cdot s_i)$$

If for all s_i, $v \cdot s_i \leqslant 0$, then it must hold that for all s_i, $v \cdot s_i = 0$, but then $\vec{0}$ is not in the interior of $\operatorname{conv}(S)$.

($\Leftarrow$) : If $\vec{0} \notin \operatorname{int}(\operatorname{conv}(S))$ then there exists a hyperplane $\mathcal{H}$ containing $\vec{0}$ such that $S \subseteq \mathcal{H}^+$. Since $\vec{0}$ is a point in the hyperplane we can write the hyperplane equation as $vx = 0$ for some $v \in \mathbb{R}^d \backslash \{\vec{0}\}$. It follows that $v \cdot s \leqslant 0$ for every $s \in S$, hence S is not complete. ◀

This leads to the following alternative statement of Steinitz's theorem.

▶ **Corollary 3.5.** *If $S \subseteq \mathbb{R}^d$ is complete, then there exists $R \subseteq S$ of size at most $2d$, such that R is complete.*

W.l.o.g., assume that $\vec{0}$ is the unique solution of P and let $\text{opt}(P) = \text{cost}(\vec{0}, P) = r$. Since $\vec{0}$ is unique, we can further assume that $\left\|\vec{0} - p\right\|_1 = r$ for every $p \in P$; otherwise, any points that do not satisfy this can be removed without affecting the optimal solution or its cost. Given a pair $a = (\sigma, p) \in \{-1, 1\}^d \times P$, we denote by $\mathcal{H}_a$ the hyperplane $\sigma \cdot (x - p) = r$ and by $\mathcal{H}_a^+$ the halfspace $\sigma \cdot (x - p) \leqslant r$. In general, $x \in \mathbb{R}^d$ is a feasible 1-center of P if for every $p \in P$, $\|x - p\|_1 \leqslant r$. Assuming r is known, we can write the LP formulation of the 1-center problem with d variables $x = (x_1, ..., x_d)$ and the $2^d|P|$ constraints $\sigma \cdot (x - p) \leqslant r$ for every $\sigma \in \{-1, 1\}^d$ and $p \in P$. That is, every constraint defines a hyperplane $\mathcal{H}_{(\sigma,p)}$. We further remove from the set of constraints $\{-1, 1\}^d \times P$, pairs (σ, p) for which $\vec{0} \notin \mathcal{H}_{(\sigma,p)}$ and denote the updated set of constraint by $A = S \times P$. Note that no points from P are removed in this process, as each has distance r from $\vec{0}$. Since $\vec{0}$ is the unique optimal solution of P, this also does not change the set of feasible solutions, and $\vec{0}$ is the unique intersection point of all hyperplanes in A.

▶ **Proposition 3.6.** *Let $A = S \times P$ be non-empty set such that $\vec{0} \in \mathcal{H}_a$ for every $a \in A$. Then, $\bigcap_{a \in A} \mathcal{H}_a^+ = \{\vec{0}\}$ if and only if S is complete.*

Proof. ($\Rightarrow$) : Let $v \in \mathbb{R}^d \backslash \{\vec{0}\}$, then there exists $a \in A$ such that $v \notin \mathcal{H}_a^+$. That is, $\sigma \cdot (v - p) > r$ and since $\vec{0} \in \mathcal{H}_a$, $\sigma \cdot (\vec{0} - p) = r$, and it follows $\sigma \cdot v > 0$.

($\Leftarrow$) : A is non-empty so clearly $\vec{0} \in \bigcap_{a \in A} \mathcal{H}_a^+$. Let $v \neq \vec{0}$, then there exists $(\sigma, p) \in A$ such that $\sigma \cdot v > 0$. Since $\vec{0} \in \mathcal{H}_a$ we have $\sigma(\vec{0} - p) = r$, it follows that $\sigma(v - p) > r$, that is, $v \notin \mathcal{H}_a^+$. ◀

We are now ready to prove our main Theorem:

▶ **Theorem 3.7.** *Every instance $P \subset \mathbb{R}^d$ with unique optimum admits a weak 0-coreset of size $2d$.*

Proof. By Proposition 3.6 the set S is complete and by Corollary 3.5 there exists $R \subseteq S$ such that R is complete and of size at most $2d$. For each $\sigma \in R$ we select a single point $p \in P$ such that the pair (σ, p) is a constraint in A. That is we obtain a set of at most $2d$ different points from P we denote by Q and denote by A' this refined set of constraints. Since R is complete, using the second direction of Proposition 3.6 we have that $\bigcap_{a \in A'} \mathcal{H}_a^+ = \{\vec{0}\}$, meaning that Q is a weak 0-coreset of size at most $2d$. ◀

We conclude this section by providing an example of a set P of size $2d$, where P is the sole weak 0-coreset of itself, demonstrating that the upper bound of $2d$ cannot be further improved. The set P is composed of the row vectors of the Hadamard matrix and their negations. The proof is provided in Appendix A.1.

4 Value-preserving coresets

In this section we provide an algorithm for constructing a dimension-independent value-preserving ϵ-coreset for 1-center. In addition, we provide a complete characterization for the special case $\epsilon = 0$, by showing a 0-coreset of size $d + 1$, and moreover that this bound is tight, see in Section 4.2.

▶ **Theorem 4.1.** *For every instance $P \subset \mathbb{R}^d$ and $0 < \epsilon < 1$, there exists a subset $S \subseteq P$ of $O(\frac{1}{\epsilon^2} \log \frac{1}{\epsilon})$ points such that*

$$\text{opt}(S) \geqslant (1 - \varepsilon)\,\text{opt}(P).$$

Our proof of Theorem 4.1 is based on the following LP formulation of 1-center and its dual. Unlike many primal-dual algorithms, our algorithm needs to explicitly solve and use this optimal dual solution to guide the coreset construction.

LP formulation of 1-center. Observe that 1-center on input $P \subset \mathbb{R}^d$ is equivalent to the following mathematical program with variables r and $x = (x_1, \ldots, x_d)$.

$$\begin{aligned} \text{minimize} \quad & r \\ \text{subject to} \quad & \sum_{i\in[d]} |p_i - x_i| \leqslant r && \forall p \in P. \end{aligned}$$

This is formally not an LP due to the absolute values in the constraints, but each such constraint can obviously be expanded into 2^d linear constraints, which leads to the following equivalent LP formulation.

$$\begin{aligned} \text{minimize} \quad & r \\ \text{subject to} \quad & \sum_{i\in[d]} \sigma_i(p_i - x_i) \leqslant r && \forall \sigma \in \{\pm 1\}^d, p \in P \\ & r \in \mathbb{R}, x \in \mathbb{R}^d. \end{aligned}$$

We derive the dual of this LP, with $2^d n$ variables $\{u_{\sigma,p}\}_{\sigma,p}$, as follows. One can think of each pair (σ, p) as generating a "new" point $(\sigma_1 p_i, \ldots, \sigma_d p_d) \in \mathbb{R}^d$, that is obtained from $p \in P$ by flipping signs.

$$\begin{aligned} \text{maximize} \quad & \sum_{\sigma\in\{\pm1\}^d} \sum_{p\in P} \sum_{i\in[d]} \sigma_i p_i u_{\sigma,p} && && \text{(DLP)} \\ \text{subject to} \quad & \sum_{\sigma\in\{\pm1\}^d} \sum_{p\in P} \sigma_i u_{\sigma,p} = 0 && \forall i \in [d] \\ & \sum_{\sigma\in\{\pm1\}^d} \sum_{p\in P} u_{\sigma,p} = 1 \\ & u_{\sigma,p} \geqslant 0 && \forall \sigma \in \{\pm 1\}^d, p \in P. \end{aligned}$$

Coreset Algorithm. Our coreset construction, presented in Algorithm 1, builds the coreset by sampling, where the probabilities come from an optimal solution to (DLP). Technically, our sampling step is similar to randomized rounding that is often used in approximation algorithms, however we use it here to find a coreset rather than an approximate solution. The analysis of this algorithm appears in Section 4.1. We slightly abuse terminology and refer to $\widehat{W}$ as a multiset, but formally it is always a sequence of m pairs, namely, $\widehat{W} := ((\hat{\sigma}^1, p^1), \ldots, (\hat{\sigma}^m, p^m))$, hence, summing over all $(\hat{\sigma}, p) \in \widehat{W}$ actually means summation over $j \in [m]$.

Algorithm 1 Value-preserving ϵ-coreset.

1 shift P such that $\sum_{p\in P} p = \vec{0}$ and solve (DLP) on P to obtain an optimal solution u^*
2 let $\widehat{W}$ be a multiset of $m := O(\frac{1}{\epsilon^2} \log \frac{1}{\epsilon})$ i.i.d. samples from the distribution u^*
    // m is an even number, u^* viewed as distribution over $\{\pm1\}^d \times P$
3 return $S := \{p : (\sigma, p) \in \widehat{W}\}$

▶ Remark 4.2. Algorithm 1 can be implemented in poly(nd)-time. The most expensive step is clearly to solve the dual LP. Since it has only $d+1$ constraints, one can solve it while using only poly(d) non-zero variables. Alternatively, one can first solve the primal LP using an ellipsoid algorithm, to find a subset of poly(n) constraints that has the same optimal value, and then solve the dual of this smaller primal LP. Moreover, inspecting the analysis, particularly (6), one can verify that it suffices to $(1+\varepsilon)$-approximate the dual LP.

4.1 Proof of Theorem 4.1: Analysis of Algorithm 1

We shall prove that the algorithm's output S satisfies $\Pr[\mathrm{opt}(S) \geqslant (1-\epsilon)\,\mathrm{opt}(P)] \geqslant 0.8$. The plan is to build a dual solution for $S \subset \mathbb{R}^d$, and show that with high probability, this dual solution is feasible for (DLP) and its objective is at least $(1-\epsilon)$ times the optimal value (of the same dual LP) for P. This would imply that $\mathrm{DLP}(S) \geqslant (1-\epsilon)\cdot\mathrm{DLP}(P)$, where $\mathrm{DLP}(P)$ denotes the optimal value of (DLP) on input P, and by strong LP duality this is equivalent to $\mathrm{opt}(S) \geqslant (1-\epsilon)\cdot\mathrm{opt}(P)$, proving the theorem.

A lower bound on opt(P). Recall that Algorithm 1 shifts P so that $\sum_{p\in P} p = \vec{0}$. This does not change the optimal value, hence our analysis simply assumes that the input P already satisfies $\sum_{p\in P} p = \vec{0}$, avoiding cumbersome notations for before and after the shift. This property of P yields the following immediate lower bound on $\mathrm{opt}(P)$.

▶ **Lemma 4.3.** *If $P \subset \mathbb{R}^d$ satisfies $\sum_{p\in P} p = \vec{0}$, then $\mathrm{opt}(P) \geqslant \max\{\|p\|_1/2 :\ p \in P\}$.*

Proof. For every point $p \in P$,

$$\|p\|_1 = \left\| \tfrac{1}{|P|} \sum_{q\in P} (p-q) \right\|_1 \leqslant \tfrac{1}{|P|} \sum_{q\in P} \|p-q\|_1 \leqslant 2\,\mathrm{opt}(P). \qquad \blacktriangleleft$$

Dual solution induced by $\widehat{W}$. We now wish to use $\widehat{W}$ from Algorithm 1 to construct a solution for the dual LP. Informally, the idea is to construct a solution $u^{\widehat{W}}$ (can also think of it as a vector with $2^d n$ coordinates), that is the average of m sparse solutions, namely, each has a single non-zero variable, whose value is 1 and it is drawn at random from the probability distribution defined by u^* (think of u^* as a vector with $2^d n$ coordinates that sum to 1), using precisely the random draws made in Algorithm 1. Next, we formalize this construction in a slightly more general form.

Given a multiset $Z \subseteq \{\pm 1\}^d \times P$ of size m, such as $\widehat{W}$, we define the *dual solution induced* by Z, denoted u^Z, as follows. Take each pair $(\sigma, p) \in Z$ and build a dual solution u, where the variable corresponding to this pair is set as $u_{\sigma,p} = 1$, and the other $2^d n - 1$ variables are set to zero, and then we average all these m solutions. Formally, u^Z is given by

$$u^Z_{\sigma',p'} := \tfrac{1}{m} \sum_{(\sigma,p)\in Z} \mathbb{1}_{\{\sigma'=\sigma, p'=p\}}.$$

The following fact rewrites certain linear forms over u^Z, that appear in (DLP), in terms of pairs $(\sigma, p) \in Z$.

▶ **Fact 4.4.** *Let u^Z be induced by some $Z \subseteq \{\pm 1\}^d \times P$ as above. Then for every $i \in [d]$,*

$$\sum_{\sigma\in\{\pm1\}^d} \sum_{p\in P} \sigma_i u^Z_{\sigma,p} = \tfrac{1}{m} \sum_{(\sigma,p)\in Z} \sigma_i,$$
$$\sum_{\sigma\in\{\pm1\}^d} \sum_{p\in P} \sigma_i p_i u^Z_{\sigma,p} = \tfrac{1}{m} \sum_{(\sigma,p)\in Z} \sigma_i p_i.$$

Initial dual solution. Before building a *feasible* dual solution for (DLP) on S, we first consider an initial dual solution $\widehat{u} := u^{\widehat{W}}$, which is just the solution induced by $\widehat{W}$ from Algorithm 1. We shall see below that $\widehat{u}$ is feasible in an expected sense, and that turning the expectation into a high-probability guarantee introduces an additive error in the constraints (and in the objective). Nonetheless, this is still useful as we will later "fix" this solution into one that does satisfy the constraints (without additive error).

Let us examine this initial random solution $\widehat{u}$. It always satisfies the second constraint of (DLP) by construction (recall that $\widehat{u}$ is the average of m sparse solutions), i.e.,

$$\sum_{\sigma\in\{\pm1\}^d}\sum_{p\in S}\widehat{u}_{\sigma,p} = m\cdot\tfrac{1}{m} = 1.$$

By linearity of expectation, $\widehat{u}$ satisfies the first constraint in expectation (the middle term uses Theorem 4.4 to provide an alternate formulation),

$$\forall i\in[d],\quad \mathbb{E}\Big[\sum_{\sigma\in\{\pm1\}^d}\sum_{p\in P}\sigma_i\widehat{u}_{\sigma,p}\Big] = \mathbb{E}\Big[\tfrac{1}{m}\sum_{(\sigma,p)\in\widehat{W}}\sigma_i\Big] = \tfrac{1}{m}\cdot m\cdot\sum_{\sigma\in\{\pm1\}^d}\sum_{p\in P}\sigma_i u^*_{\sigma,p} = 0. \quad (5)$$

And again by linearity of expectation, the expected objective of $\widehat{u}$ is the same as the optimal solution u^*, because:

$$\mathbb{E}\Big[\sum_{\sigma\in\{\pm1\}^d}\sum_{p\in P}\sum_{i\in[d]}\sigma_i p_i\widehat{u}_{\sigma,p}\Big] = \mathbb{E}\Big[\frac{1}{m}\sum_{i\in[d]}\sum_{(\sigma,p)\in\widehat{W}}\sigma_i p_i\Big] = \mathrm{opt}(P). \quad (6)$$

Almost feasibility. We bound the deviation of $\widehat{u}$ from satisfying the first constraint, as follows. Consider $i\in[d]$ and let $\mathcal{F}_i$ be the event that $|\sum_{\sigma\in\{\pm1\}^d}\sum_{p\in S}\sigma_i\widehat{u}_{\sigma,p}| > \epsilon$, which by Theorem 4.4 can be also written as $|\frac{1}{m}\sum_{(\sigma,p)\in\widehat{W}}\sigma_i| > \epsilon$. Informally, we want to bound the probability of this event, because when it does not occur, the constraint is "almost satisfied". By applying Hoeffding's inequality, where we use (5) for the expectation and the fact that $|\sigma_i|\leqslant 1$, we get (by our choice of m),

$$\Pr[\mathcal{F}_i] = \Pr\Big[\Big|\tfrac{1}{m}\sum_{(\sigma,p)\in\widehat{W}}\sigma_i\Big| > \epsilon\Big] \leqslant 2e^{-\Omega(\epsilon^2 m)} \leqslant \epsilon. \quad (7)$$

A feasible solution u. We next use $\widehat{W}$ to construct a new multiset W whose induced dual u^W is feasible, and in particular satisfies the first constraint. We will then take our final dual solution to be $u := u^W$, and it will remain to bound its objective value. At a high level, W is obtained by "flipping" some of the signs appearing in $\widehat{W}$. More precisely, given $\widehat{W} = ((\widehat{\sigma}^1,p^1),\ldots,(\widehat{\sigma}^m,p^m))$, we shall define new sign vectors $\sigma^1,\ldots,\sigma^m$ but keep the exact same points $p^1,\ldots,p^m\in P$, and construct $W = ((\sigma^1,p^1),\ldots,(\sigma^m,p^m))$. It will be convenient to arrange the sign vectors $\widehat{\sigma}^1,\ldots,\widehat{\sigma}^m$ as the rows of a matrix $\widehat{M}\in\{\pm1\}^{m\times d}$, and use this $\widehat{M}$ to define a new matrix M, whose rows define the new sign vectors $\sigma^1,\ldots,\sigma^m$.

We construct M (from $\widehat{M}$) using the following procedure, which operates separately on each column $i\in[d]$. First copy column i of $\widehat{M}$ to be also column i of M, and let n_i^+ and n_i^- be the number of positive and negative entries in this column, respectively. If $n_i^+ = n_i^-$, leave this column of M as is, noticing that it sums to zero. Otherwise, flip $\frac{|n_i^+ - n_i^-|}{2}$ signs chosen carefully from among the majority sign in column i, and this will ensure that column i of M sums to zero. (This is possible because m is even.) The careful choice (which signs to flip) will favor row indices $j\in[m]$ with small value $|p_i^j|$. Formally, let $R_i\subseteq[m]$ be the

set of rows j with the majority sign in column i of $\widehat{M}$, then sort the indices $j \in R_i$ by their value $|p_i^j|$, pick the $|n_i^+ - n_i^-|/2$ indices with smallest value (breaking ties arbitrarily), and flip their entries. As explained above, the rows of the resulting matrix M define sign vectors $\sigma^1, \ldots, \sigma^m$, which in turn define $W = ((\sigma^1, p^1), \ldots, (\sigma^m, p^m))$, and our final dual solution is $u = u^W$.

▶ **Lemma 4.5.** *The solution $u = u^W$ is feasible for* (DLP) *on S.*

Proof. The first constraint is satisfied because the signs in every column of M sum to zero. The second constraint is satisfied because flipping a sign σ_i^j "moves" some amount (say $1/m$) from variable $u_{\sigma,p}$ to $u_{\sigma',p}$, where σ, σ' differ in coordinate j, but the total remains the same. Finally, it uses only variables $u_{\sigma,p}$ for $p \in S$, i.e., all other variables are 0, and thus it is feasible not only for (DLP) on P but also on S. ◀

The next lemma bounds the decrease in objective when constructing w from $\widehat{W}$, i.e., comparing that of u with that of $\widehat{u}$. Its proof is based on an averaging argument.

▶ **Lemma 4.6.** *For all $i \in [d]$,*

$$\Big| \sum_{(\sigma,p)\in\widehat{W}} \sigma_i p_i - \sum_{(\sigma,p)\in W} \sigma_i p_i \Big| \leqslant \frac{2|n_i^+ - n_i^-|}{m} \sum_{(\sigma,p)\in\widehat{W}} |p_i|.$$

Proof. We consider only the case $n_i^+ \geqslant n_i^-$, as the other case is symmetric. Since $n_i^+ + n_i^- = m$, we have $n_i^+ \geqslant m/2$. Recall that the construction of M flips in column i exactly $\frac{n_i^+ - n_i^-}{2}$ signs, picking the indices $j \in R_i$ with smallest value $|p_i^j|$. Every such flip changes the objective by $2|p_i^j|$, and the number of choices $|R_i| = n_i^+ \geqslant m/2$, hence by an averaging argument,

$$\Big| \sum_{(\sigma,p)\in\widehat{W}} \sigma_i p_i - \sum_{(\sigma,p)\in W} \sigma_i p_i \Big| \leqslant \frac{(n_i^+ - n_i^-)/2}{|R_i|} \sum_{j\in R_i} 2|p_i^j| \leqslant \frac{n_i^+ - n_i^-}{m/2} \sum_{j\in[m]} |p_i^j|. \qquad ◀$$

The objective value of u. Using Theorem 4.4, we can write the objective value of u as

$$L := \sum_{i\in[d]} \sum_{\sigma\in\{\pm1\}^d} \sum_{p\in S} \sigma_i p_i u_{\sigma,p} = \tfrac{1}{m} \sum_{i\in[d]} \sum_{(\sigma,p)\in W} \sigma_i p_i. \tag{8}$$

and that of $\widehat{u}$ as

$$\widehat{L} := \sum_{i\in[d]} \sum_{\sigma\in\{\pm1\}^d} \sum_{p\in S} \sigma_i p_i \widehat{u}_{\sigma,p} = \tfrac{1}{m} \sum_{i\in[d]} \sum_{(\sigma,p)\in\widehat{W}} \sigma_i p_i, \tag{9}$$

Our plan is to bound L (with high probability) by relating it to $\widehat{L}$, and then bound the latter using Chebyshev's inequality. These steps rely on the next two lemmas, and before proving them, we show how they imply Theorem 4.1.

▶ **Lemma 4.7.** $\mathbb{E}[|L - \widehat{L}|] \leqslant O(\epsilon)\operatorname{opt}(P)$.

▶ **Lemma 4.8.** $\operatorname{Var}(\widehat{L}) \leqslant \frac{4}{m}(\operatorname{opt}(P))^2$.

Proof of Theorem 4.1. We already established in Theorem 4.5 that u is a feasible solution for (DLP) on S, hence $\operatorname{opt}(S) \geqslant L$. It thus suffices to show that $\Pr[L \geqslant (1 - O(\epsilon))\operatorname{opt}(P)] \geqslant 0.8$. By Theorem 4.7 and Markov's inequality, $\Pr[|L - \widehat{L}| \leqslant O(\epsilon)\operatorname{opt}(P)] \geqslant 0.9$. Next, we apply Chebyshev's inequality to $\widehat{L}$, using the variance bound Theorem 4.8 and our choice of m, to obtain $\Pr[|\widehat{L} - \mathbb{E}[\widehat{L}]| \geqslant \epsilon \operatorname{opt}(P)] \leqslant \frac{4/m}{\epsilon^2} \leqslant 0.1$. Recall also from (6) that $\mathbb{E}[\widehat{L}] = \operatorname{opt}(P)$. It follows by a union bound that with probability at least 0.8,

$$L \geqslant \widehat{L} - O(\epsilon)\operatorname{opt}(P) \geqslant (1 - O(\epsilon))\operatorname{opt}(P). \qquad ◀$$

Proof of Theorem 4.7. Observe that $\mathcal{F}_i$ is the event that $|n_i^+ - n_i^-| > \epsilon m$, in which case we can still bound $|n_i^+ - n_i^-| \leqslant m$ (which holds always). Thus, $|n_i^+ - n_i^-| \leqslant \epsilon m + \mathbb{1}_{\mathcal{F}_i} \cdot m$. Now applying Theorem 4.6 for every $i \in [d]$ and taking a sum, we get (after dividing by m)

$$\begin{aligned}\tfrac{1}{m}\Big|\sum_{i\in[d]}\sum_{(\sigma,p)\in\widehat{W}}\sigma_i p_i - \sum_{i\in[d]}\sum_{(\sigma,p)\in W}\sigma_i p_i\Big| &\leqslant \tfrac{1}{m}\sum_{i\in[d]}\frac{2|n_i^+ - n_i^-|}{m}\sum_{(\sigma,p)\in\widehat{W}}|p_i| \\ &\leqslant \tfrac{2}{m}\sum_{i\in[d]}\Big[(\epsilon + \mathbb{1}_{\mathcal{F}_i})\sum_{(\sigma,p)\in\widehat{W}}|p_i|\Big]. \qquad (10)\end{aligned}$$

To bound (10) further, we expand its final expression into two parts. We then bound one part in (11) using Theorem 4.3, and the other part in Theorem 4.9.

$$\mathbb{E}\Big[\tfrac{1}{m}\sum_{i\in[d]}\epsilon\sum_{(\sigma,p)\in\widehat{W}}|p_i|\Big] = \epsilon\cdot\mathbb{E}\Big[\tfrac{1}{m}\sum_{(\sigma,p)\in\widehat{W}}\|p_i\|_1\Big] \leqslant 2\epsilon\cdot\mathrm{opt}(P). \qquad (11)$$

▷ Claim 4.9. $\mathbb{E}\Big[\tfrac{1}{m}\sum_{i\in[d]}\mathbb{1}_{\mathcal{F}_i}\sum_{(\sigma,p)\in\widehat{W}}|p_i|\Big] \leqslant 2\epsilon\cdot\mathrm{opt}(P).$

Proof. The main obstacle here is that we have a product of two random variables, $\mathbb{1}_{\mathcal{F}_i}$ and $|p_i|$, that are not independent. Intuitively, their dependence should be relatively small, and our actual proof bounds their product with another product, of two independent random variables. Recall that $\widehat{W} = \{(\hat{\sigma}^1, p^1), \ldots, (\hat{\sigma}^m, p^m)\}$ consists of m i.i.d. samples, hence we can focus on analyzing (say) the last one, formally

$$\mathbb{E}\Big[\tfrac{1}{m}\sum_{i\in[d]}\mathbb{1}_{\mathcal{F}_i}\cdot\sum_{(\sigma,p)\in\widehat{W}}|p_i|\Big] = \mathbb{E}\Big[\sum_{i\in[d]}\mathbb{1}_{\mathcal{F}_i}\cdot|p_i^m|\Big].$$

Consider $i \in [d]$, and define $\mathcal{F}_i'$ to be the event that $|\frac{1}{m}\sum_{j=1}^{m-1}\hat{\sigma}_i^j| > \epsilon - \frac{1}{m}$. Observe that $\mathbb{1}_{\mathcal{F}_i} \leqslant \mathbb{1}_{\mathcal{F}_i'}$ because if $\mathcal{F}_i$ occurs then also $\mathcal{F}_i'$ must occur. The crux is that $\mathcal{F}_i'$ is independent of $(\hat{\sigma}^m, p^m)$, and thus $\mathbb{E}[\mathbb{1}_{\mathcal{F}_i'}\cdot|p_i^m|] = \mathbb{E}[\mathbb{1}_{\mathcal{F}_i'}]\cdot\mathbb{E}[|p_i^m|]$. We can still bound $\mathbb{E}[\mathbb{1}_{\mathcal{F}_i'}] = \Pr[\mathcal{F}_i'] \leqslant \epsilon$ by the argument we had for $\mathcal{F}_i$ in (7), except that we use $m-1$ instead of m. Altogether,

$$\mathbb{E}\Big[\sum_{i\in[d]}\mathbb{1}_{\mathcal{F}_i}\cdot|p_i^m|\Big] \leqslant \mathbb{E}\Big[\sum_{i\in[d]}\mathbb{1}_{\mathcal{F}_i'}\cdot|p_i^m|\Big] = \sum_{i\in[d]}\mathbb{E}[\mathbb{1}_{\mathcal{F}_i'}]\cdot\mathbb{E}[|p_i^m|] \leqslant \epsilon\cdot\mathbb{E}[\|p^m\|_1],$$

and the claim follows by using Theorem 4.3 to bound $\|p^m\|_1 \leqslant 2\,\mathrm{opt}(P)$. ◁

We now proceed with the proof of Theorem 4.7. Plugging the bounds from (11) and Theorem 4.9 into (10), we obtain

$$\mathbb{E}\Big[\tfrac{1}{m}\Big|\sum_{i\in[d]}\sum_{(\sigma,p)\in W}\sigma_i p_i - \sum_{i\in[d]}\sum_{(\sigma,p)\in\widehat{W}}\sigma_i p_i\Big|\Big] \leqslant O(\epsilon)\,\mathrm{opt}(P).$$

which completes the proof of Theorem 4.7. ◀

Proof of Theorem 4.8. For $\sigma \in \{\pm 1\}^d$ and $p \in P$, define $X_{\sigma,p} := \sum_{i\in[d]}\sigma_i p_i$. Then we can write

$$\widehat{L} = \tfrac{1}{m}\sum_{i\in[d]}\sum_{(\sigma,p)\in\widehat{W}}\sigma_i p_i = \tfrac{1}{m}\sum_{(\sigma,p)\in\widehat{W}}X_{\sigma,p}.$$

Observe that for every (σ, p) we have $|X_{\sigma,p}| \leqslant \sum_{i\in[d]} |p_i| \leqslant 2\operatorname{opt}(P)$ by Theorem 4.3. Now since the m samples in $\widehat{W}$ are independent, the variance of their sum is the sum of their variances, and we obtain

$$\operatorname{Var}(\widehat{L}) = \tfrac{1}{m^2} \cdot \operatorname{Var}\Big(\sum_{(\sigma,p)\in\widehat{W}} X_{\sigma,p}\Big) \leqslant \tfrac{1}{m^2} \cdot m \cdot (2\operatorname{opt}(P))^2 = \tfrac{4}{m}(\operatorname{opt}(P))^2. \tag{12}$$

This completes the proof of Theorem 4.8. ◀

4.2 Lower bound for value-preserving 0-coreset

The Helly number of a body S, denoted $\mathcal{H}(S)$, is the smallest positive integer h, such that if every subset of size h from a family of translations of S has a non-empty intersection, then the entire family also has a non-empty intersection. The Helly theorem shows that the Helly number of every convex body is smaller than or equal to $d + 1$.

The Helly number is equivalent to a worst-case result of value-preserving 0-coreset (cf. [6]). In the next proposition we show that the $d + 1$ upper bound is tight, that is, the Helly number of the ℓ_1 ball is $d + 1$.

▶ **Proposition 4.10.** *Assume $d > 2$. There exists $P \subset \mathbb{R}^d$, such that every value-preserving 0-coreset must have size $d + 1$.*

Proof. Consider the set of $d + 1$ points $P = \{p_0, \ldots, p_d\}$, where $p_0 = (-1, -1, \ldots, -1)$ and p_i is the vector with all entries equal to 1 except for a -1 in the i-th position. We will show that:

- The point $\vec{0} = (0, 0, \ldots, 0)$ is the only center of P with a cost of d.
- Each subset of P containing d points has a cost strictly less than d.

To prove the first claim, let c be an optimal center of P, clearly $c \in [-1, 1]^d$. We know that for every i:

$$\operatorname{cost}(c, P) \geqslant \frac{1}{2}(\|c - p_0\|_1 + \|c - p_i\|_1) = \frac{1}{2}(2(d-1) + 2|c_i + 1|) = d - 1 + |c_i + 1|$$

On the other hand, $\operatorname{cost}(c, P) \leqslant d$, since the zero vector $\vec{0}$ has a cost d, consequently $1 \geqslant |c_i + 1|$ for all i. This implies $c_i \leqslant 0$ for all i, leading to:

$$\operatorname{cost}(c, P) \geqslant \|c - p_i\|_1 \geqslant (d-1) + (1 + c_i) - \sum_{j\neq i} c_j = d + c_i - \sum_{j\neq i} c_j$$

Since $d \geqslant \operatorname{cost}(c, P)$, it follows that $\sum_{j\neq i} c_j \geqslant c_i$ for all i. Given that $d > 2$, this implies $c = \vec{0}$.

For the second claim, consider any subset Q of d points. If $p_0 \notin Q$, then clearly $\operatorname{opt}(Q) < d$. Assuming $p_0 \in Q$ and, without loss of generality, $p_d \notin Q$, we set $c_i = -\frac{1}{d-2}$ for $i < d$, and $c_d = 1$. The cost for this choice of c is:

$$\begin{aligned}\operatorname{cost}(c, P) &= \max\Big\{2 + (d-1)|-1-c_i|, (d-2)|1-c_i| + |-1-c_i|\Big\} \\ &= \max\Big\{2 + (d-1)(1+c_i), (d-2)(1-c_i) + (1+c_i)\Big\} < d\end{aligned}$$ ◀

5 Generalization to discrete metric spaces

In this section we show how to extend our lower bounds results to other discrete metric spaces by providing two examples for Jaccard and Kendall's tau metric spaces.

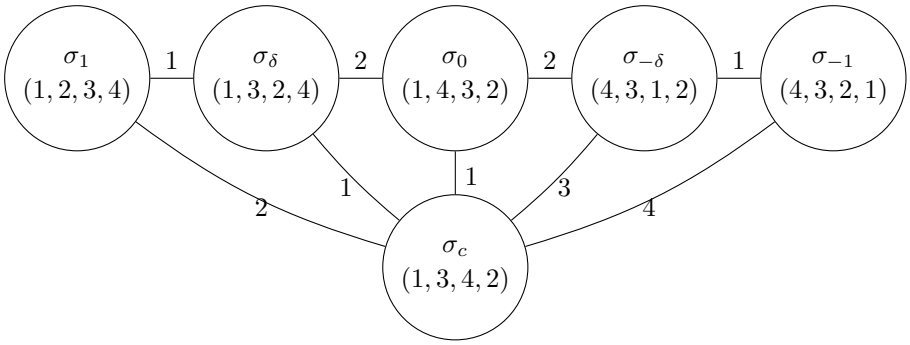


Figure 1 The figure shows the distances between the different base permutations used in the construction of the set P in Proposition 5.1. The distances between $\sigma_1, \sigma_\delta, \sigma_0, \sigma_{-\delta}, \sigma_{-1}$ are additive over the straight lines. The curved lines indicate the distances from $(1,3,4,2)$ to the others.

5.1 Theorem 2.2 in Kendall's tau metric space

Denote by $\mathcal{S}_d$ the set of permutations over the set of elements $[d]$. For two permutations $\sigma_1, \sigma_2 \in \mathcal{S}_d$, the Kendall's tau distance is the number of pairs (i,j) such that the order of i and j is reversed between σ_1 and σ_2. Formally,

$$\tau(\sigma_1,\sigma_2) = \left|\left\{\{i,j\} : (\sigma_1(i)-\sigma_1(j))(\sigma_2(i)-\sigma_2(j)) < 0\right\}\right|$$

Following a similar approach as in Section 2 we will prove a lower bound of size $2^{\Omega(d)}$ on the size of weak coreset.

▶ **Proposition 5.1.** *There exists a set $P \subseteq \mathcal{S}_{4d}$, such that for all $\epsilon \in [0, \frac{1}{3})$, if $Q \subseteq P$ is a weak ϵ-coreset of P, then $|Q| \geqslant 2^{\Omega(d)}$.*

Proof. We will design a gadget that will allow us to replicate the structure of P that is given in the proof of Theorem 2.2. We denote $\sigma_1 = (1,2,3,4)$, $\sigma_\delta = (1,3,2,4)$, $\sigma_0 = (1,4,3,2)$, $\sigma_{-\delta} = (4,3,1,2)$, $\sigma_{-1} = (4,3,2,1)$ and $\sigma_c = (1,3,4,2)$. The distances between these base permutations are detailed in Figure 1.

We will now introduce some notations that will allow us to describe permutations in $\mathcal{S}_{4d}$ using the base permutations. For $\sigma = (\pi_1, \ldots, \pi_k)$, with a slight abuse of notation, we denote by $i + \sigma$ the permutation $(i + \pi_1, \ldots, i + \pi_k)$ and also introduce an operator $\odot$ to denote the product $A \odot B = \{\sigma \cdot \hat{\sigma} : \sigma \in A, \hat{\sigma} \in B\}$ where the product of two permutations is their composition. We will extend a base permutation over 4 elements to a permutation over $4d$ elements using the notation $\sigma_{\vec{x}} = (0 + \sigma_x) \cdot (4 + \sigma_x) \cdot \ldots \cdot (4(d-1) + \sigma_x)$, for $x \in \{-1, -\delta, 0, \delta, 1, c\}$. Now we are ready to describe P. Let $B = \odot_{i=0}^{d-1}\{4i + \sigma_\delta, 4i + \sigma_{-\delta}\}$, then $P = B \cup \{\sigma_{\vec{1}}, \sigma_{-\vec{1}}\}$. The permutation $\sigma_{\vec{0}}$, has cost $3d$ (see also Figure 1) and this cost is optimal the base permutations composing $\sigma_{\vec{1}}, \sigma_{-\vec{1}}$ are antipodal (for every $i \leqslant d-1$, every pair that agrees with $(4i + \sigma_1)$ disagrees with $(4i + \sigma_{-1})$, and vice versa).

Let Q be a weak ϵ-coreset of P. Notice that Q must contain both $\sigma_{-\vec{1}}$ and $\sigma_{\vec{1}}$, as otherwise, if $\sigma_{-\vec{1}}, \sigma_{\vec{1}} \notin Q$, consider $\sigma_{\vec{0}}$ as a center, and if Q contains just one of $\{\sigma_{-\vec{1}}, \sigma_{\vec{1}}\}$ say $\sigma_{\vec{1}}$, consider the permutation $\sigma_{\vec{c}}$ as a center. In both cases, we obtain that $2d = \mathrm{opt}(Q) < \frac{1}{1+\epsilon}\,\mathrm{opt}(P) = \frac{1}{1+\epsilon}3d$, in contradiction that Q is a weak ϵ-coreset.

Similarly to Proposition 2.3 it follows that if $|Q| \leqslant \frac{1}{\sqrt{2d}}2^{0.18d}$, then there exists $b \in B$ such that the Kendall tau distance $\tau(b,q) \leqslant 3d$ for every $q \in Q$. That is, b is an optimal center of Q. However, $\mathrm{cost}(b,P) = 4d$, by considering the opposite permutation of b in P, thus $|Q| \geqslant 2^{\Omega(d)}$. ◀

5.2 Theorem 3.1 in the Jaccard metric space

For two sets A and B, the Jaccard distance is defined as: $J(A,B) = 1 - \frac{|A\cap B|}{|A\cup B|}$. We will use the symbol $\circledast$ to denote the product $A \circledast B = \{a \cup b : a \in A, b \in B\}$.

Following a similar approach as in Section 3.1 we will prove a lower bound of size $\Omega(\log d)$ on the size of weak coreset.

▶ **Proposition 5.2.** *Let $P = \circledast_{i=0}^{d-1} \{\{3i\}, \{3i, 3i+1, 3i+2\}\}$ be a collection of sets, i.e. points in the Jaccard metric space. For every $\epsilon < \frac{1}{3}$, a weak ϵ-coreset has $\Omega(\log d)$ points.*

Proof. Let $\mathcal{U} = \{0, 1, \ldots, 3d-1\}$ denote the universe. We also denote elements in $\mathcal{U}$ as *type 0* if their remainder modulo 3 is 0 and *type 1* otherwise. Note that all type 0 elements appear in every subset of P and thus appear in every optimal center of every coreset of P. It is then easy to verify that the set of optimal centers of P is the $\circledast$ product of $\{3i+1, 3i+2\}$ for $i \in [d]$, along with all type 0 elements, and that $\mathrm{opt}(P) = \frac{1}{2}$.

Using the same arguments as in the proof of Theorem 3.1, assume $Q \subset P$ is a weak ϵ-coreset of size $t \leqslant \frac{1}{2}\log d$. We denote by $A_i = \{3i\}$ and $B_i = \{3i, 3i+1, 3i+2\}$ the two building blocks of the set P, corresponding to -1 and 1 in the proof of Theorem 3.1. Every set in $p \in Q$ induces a partition of $\mathcal{U}$ into at most two parts, depending on their assignment to A_i or B_i. Again, we use Π to denote the common refinement of all these partitions.

Let c^* be any optimal 1-center of Q, and consider a partition $A \in \Pi$. Let k denote the number of type 1 elements from A in c^*. By selecting k different type 1 elements from A, say the first k type 1 indices of A, we preserve the cost of c^* over Q, as the number of differences c^* has with each $p \in Q$ remains unchanged. We apply this modifications for all partitions in Π and denote the new center by $\bar{c}^*$. However, by the construction of $\bar{c}^*$ there exists $p \in P$ such that p intersects at most one type 1 element of $\bar{c}^*$ in every partition and the union of p and c^* is $\mathcal{U}$. Consequently, the distance between c^* and p is at least $1 - \frac{d+|\Pi|}{3d} \geqslant \frac{2}{3} - \frac{2^t}{3d} = \frac{2}{3} - \frac{\sqrt{d}}{3d} > (1+\epsilon)\,\mathrm{opt}(P)$. ◀

References

1. Amir Abboud, MohammadHossein Bateni, Vincent Cohen-Addad, Karthik C. S., and Saeed Seddighin. On complexity of 1-center in various metrics. In *APPROX/RANDOM*, volume 275 of *LIPIcs*, pages 1:1–1:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.APPROX/RANDOM.2023.1.
2. Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Approximating extent measures of points. *J. ACM*, 51(4):606–635, 2004. doi:10.1145/1008731.1008736.
3. Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Geometric approximation via coresets. In *Combinatorial and computational geometry*, volume 52 of *MSRI Publications*, chapter 1, pages 1–30. Cambridge University Press, 2005.
4. Sepideh Aghamolaei and Mohammad Ghodsi. A composable coreset for k-center in doubling metrics. In *CCCG*, pages 165–171, 2018. URL: http://www.cs.umanitoba.ca/%7Ecccg2018/papers/session4A-p2.pdf.
5. Christian Bachmaier, Franz J. Brandenburg, Andreas Gleißner, and Andreas Hofmeier. On the hardness of maximum rank aggregation problems. *J. Discrete Algorithms*, 31:2–13, 2015. doi:10.1016/j.jda.2014.10.002.
6. René Brandenberg and Stefan König. No dimension-independent core-sets for containment under homothetics. *Discrete & Computational Geometry*, 49(1):3–21, 2013. doi:10.1007/s00454-012-9462-0.
7. Mihai Bădoiu and Kenneth L. Clarkson. Optimal core-sets for balls. *Comput. Geom.*, 40(1):14–22, 2008. doi:10.1016/j.comgeo.2007.04.002.
8. Mihai Bădoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *STOC*, pages 250–257. ACM, 2002. doi:10.1145/509907.509947.

9 Marc Bury, Michele Gentili, Chris Schwiegelshohn, and Mara Sorella. Polynomial time approximation schemes for all 1-center problems on metric rational set similarities. *Algorithmica*, 83(5):1371–1392, 2021. doi:10.1007/s00453-020-00787-3.

10 Matteo Ceccarello, Andrea Pietracaprina, and Geppino Pucci. Solving k-center clustering (with outliers) in MapReduce and streaming, almost as accurately as sequentially. *Proc. VLDB Endow.*, 12(7):766–778, 2019. doi:10.14778/3317315.3317319.

11 Diptarka Chakraborty, Debarati Das, and Robert Krauthgamer. Approximating the median under the ulam metric. In *SODA*, pages 761–775. SIAM, 2021. doi:10.1137/1.9781611976465.48.

12 Diptarka Chakraborty, Debarati Das, and Robert Krauthgamer. Clustering permutations: New techniques with streaming applications. In *ITCS*, volume 251 of *LIPIcs*, pages 31:1–31:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ITCS.2023.31.

13 Diptarka Chakraborty, Kshitij Gajjar, and Agastya Vibhuti Jha. Approximating the center ranking under ulam. In *FSTTCS*, volume 213 of *LIPIcs*, pages 12:1–12:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.FSTTCS.2021.12.

14 Timothy M. Chan. Faster core-set constructions and data-stream algorithms in fixed dimensions. *Computational Geometry*, 35(1):20–35, 2006. doi:10.1016/j.comgeo.2005.10.002.

15 Ke Chen. On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009. doi:10.1137/070699007.

16 Flavio Chierichetti, Ravi Kumar, Sandeep Pandey, and Sergei Vassilvitskii. Finding the jaccard median. In *SODA*, pages 293–311. SIAM, 2010. doi:10.1137/1.9781611973075.25.

17 Michael B. Cohen, Yin Tat Lee, Gary L. Miller, Jakub Pachocki, and Aaron Sidford. Geometric median in nearly linear time. In *STOC*, pages 9–21. ACM, 2016. doi:10.1145/2897518.2897647.

18 Vincent Cohen-Addad, David Saulpic, and Chris Schwiegelshohn. A new coreset framework for clustering. In *STOC*, pages 169–182. ACM, 2021. doi:10.1145/3406325.3451022.

19 Matan Danos. Coresets for clustering by uniform sampling and generalized rank aggregation. Master's thesis, Weizmann Institute of Science, 2021.

20 Mark de Berg, Leyla Biabani, and Morteza Monemizadeh. k-center clustering with outliers in the MPC and streaming model. In *IPDPS*, pages 853–863. IEEE, 2023. doi:10.1109/IPDPS54959.2023.00090.

21 Dan Feldman. Core-sets: An updated survey. *WIREs Data Mining and Knowledge Discovery*, 10(1):e1335, 2020. doi:10.1002/widm.1335.

22 Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *STOC*, pages 569–578. ACM, 2011. doi:10.1145/1993636.1993712.

23 Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985. doi:10.1016/0304-3975(85)90224-5.

24 Peter M Gruber and Jörg M Wills. *Handbook of convex geometry, Volume B*. North-Holland, 1993.

25 Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors. In *STOC*, pages 95–103. ACM, 2007. doi:10.1145/1250790.1250806.

26 Ming Li, Bin Ma, and Lusheng Wang. On the closest string and substring problems. *J. ACM*, 49(2):157–171, 2002. doi:10.1145/506147.506150.

27 Bin Ma and Xiaoming Sun. More efficient algorithms for closest string and substring problems. *SIAM Journal on Computing*, 39(4):1432–1443, 2010. doi:10.1137/080739069.

28 Alexander Munteanu and Chris Schwiegelshohn. Coresets-methods and history: A theoreticians design pattern for approximation and streaming algorithms. *Künstliche Intell.*, 32(1):37–53, 2018. doi:10.1007/s13218-017-0519-3.

29 François Nicolas and Eric Rivals. Hardness results for the center and median string problems under the weighted and unweighted edit distances. *J. Discrete Algorithms*, 3(2-4):390–415, 2005. doi:10.1016/j.jda.2004.08.015.

30 Rina Panigrahy. Minimum enclosing polytope in high dimensions. *CoRR*, cs.CG/0407020, 2004. doi:10.48550/arXiv.CS/0407020.

31 Jeff M. Phillips. Coresets and sketches. In *Handbook of discrete and computational geometry*, chapter 48, pages 1269–1288. Chapman and Hall/CRC, 3rd edition, 2017. doi:10.1201/9781315119601.

32 Alvin Yan Hong Yao and Diptarka Chakraborty. Approximate maximum rank aggregation: Beyond the worst-case. In *FSTTCS*, volume 284 of *LIPIcs*, pages 12:1–12:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.FSTTCS.2023.12.

33 Hai Yu, Pankaj K. Agarwal, Raghunath Poreddy, and Kasturi R. Varadarajan. Practical methods for shape fitting and kinetic data structures using coresets. *Algorithmica*, 52(3):378–402, 2008. doi:10.1007/s00453-007-9067-9.

A Appendix

A.1 Lower bound for weak 0-coreset for inputs with a unique solution

We present an example of a set P of size $2d$, where P is the sole weak 0-coreset of itself. This demonstrates that the upper bound of $2d$ is indeed tight.

We start with a simple proposition regarding the distances from a given point within the unit hypercube to two antipodal vertices of the cube.

▶ **Proposition A.1.** *If $p \in \{-1,1\}^d$ and $x \in [-1,1]^d$ then $\|p-x\|_1 + \|(-p)-x\|_1 = 2d$.*

Proof. $\|p-x\|_1 + \|(-p)-x\|_1 = \sum_{i=1}^{d} |1-x_i| + |-1-x_i| = \sum_{i=1}^{d}((1-x_i)+(1+x_i)) = \sum_{i=1}^{d} 2 = 2d$ ◀

An immediate consequence of the above proposition is that the cost of every optimal center of P is at least d for every set $P \subseteq \{-1,1\}^d$ containing an antipodal pair. Since $\vec{0}$ has cost at most d to P we get that $\mathrm{opt}(P) = d$ and that $\vec{0}$ is an optimal center. This is summarized in the following corollary.

▶ **Corollary A.2.** *If $P \subseteq [-1,1]^d$ contains antipodal pair, then $\mathrm{opt}(P) = d$.*

Let H denote the set of row vectors of the Hadamard matrix, and let $P = H \cup -H$.

▶ **Proposition A.3.** *There exists $P \subset \mathbb{R}^d$ with unique optimum, such that every weak 0-coreset must have size $2d$.*

Proof. Following Corollary A.2, $\mathrm{opt}(P) = d$. We will further show that if $Q \subseteq P$ does not contain an antipodal pair, then $\mathrm{opt}(Q) < d$. Note that for $a, b \in P$, $a \neq \pm b$ we have $a \cdot b = 0$. Also, since $a, b \in \{-1,1\}^d$, $a \cdot b = \sum_{i:a_i=b_i} 1 - \sum_{i:a_i \neq b_i} 1 = d - 2\sum_{i:a_i \neq b_i} 1$. That is:

$$\|a-b\|_1 = 2 \sum_{i:a_i \neq b_i} 1 = d - a \cdot b = d \tag{13}$$

That is, the distance from every $a \in P$ to all points $P \backslash \{a, -a\}$ is d.

Let $c = \frac{1}{|Q|}\sum_{q \in Q} q$ be some center. Since Q does not contain an antipodal pair it follows that $|Q| \leqslant d$. Now, using Equation 13, for every $\tilde{q} \in Q$:

$$\|c-\tilde{q}\|_1 = \frac{1}{|Q|}\left\|\sum_{q \in Q} q - \tilde{q}\right\|_1 = \frac{1}{|Q|}\left\|\sum_{q \in Q, q \neq \tilde{q}} d\right\|_1 = \frac{|Q|-1}{|Q|}d < d$$

That is $\mathrm{opt}(Q) < d$. It follows that every weak 0-coreset of P contains an antipodal pair. If a weak 0-coreset Q is a proper subset of P then there exists $q \in Q$ with $-q \notin Q$. But $\mathrm{cost}(q, Q) = d = \mathrm{opt}(Q)$, whereas $\mathrm{cost}(q, P) = 2d$. Hence P is the sole weak 0-coreset of P. ◀

Extracting Dual Solutions via Primal Optimizers

Yair Carmon
Tel Aviv University, Israel

Arun Jambulapati
University of Michigan, Ann Arbor, MI, USA

Liam O'Carroll
Stanford University, CA, USA

Aaron Sidford
Stanford University, CA, USA

Abstract

We provide a general method to convert a "primal" black-box algorithm for solving regularized convex-concave minimax optimization problems into an algorithm for solving the associated dual maximin optimization problem. Our method adds recursive regularization over a logarithmic number of rounds where each round consists of an approximate regularized primal optimization followed by the computation of a dual best response. We apply this result to obtain new state-of-the-art runtimes for solving matrix games in specific parameter regimes, obtain improved query complexity for solving the dual of the CVaR distributionally robust optimization (DRO) problem, and recover the optimal query complexity for finding a stationary point of a convex function.

2012 ACM Subject Classification Theory of computation → Mathematical optimization

Keywords and phrases Minimax optimization, black-box optimization, matrix games, distributionally robust optimization

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.29

Related Version *Full Version*: http://arxiv.org/abs/2412.02949

Funding *Yair Carmon*: YC acknowledges support from the Israeli Science Foundation (ISF) grant no. 2486/21, and the Alon Fellowship.
Liam O'Carroll: LO acknowledges support from NSF Grant CCF-1955039.
Aaron Sidford: AS acknowledges support from a Microsoft Research Faculty Fellowship, NSF CAREER Grant CCF-1844855, NSF Grant CCF-1955039, and a PayPal research award.

1 Introduction

We consider the foundational problem of efficiently solving convex-concave games. For nonempty, closed, convex constraint sets $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \subseteq \mathbb{R}^n$ and differentiable convex-concave objective function $\psi : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$ (namely, $\psi(\cdot, y)$ is convex for any fixed y and $\psi(x, \cdot)$ is concave for any fixed x), we consider the following *primal*, minimax optimization problem (P) and its associated *dual*, maximin optimization problem (D):

$$\underset{x\in\mathcal{X}}{\text{minimize}}\, f(x) \text{ for } f(x) \coloneqq \max_{y\in\mathcal{Y}} \psi(x, y), \text{ and} \tag{P}$$

$$\underset{y\in\mathcal{Y}}{\text{maximize}}\, \phi(y) \text{ for } \phi(y) \coloneqq \min_{x\in\mathcal{X}} \psi(x, y). \tag{D}$$

If additionally $\mathcal{X}$ and $\mathcal{Y}$ are bounded (which we assume for simplicity in the introduction but generalize later), every pair of primal and dual optimizers $x^\star \in \operatorname{argmin}_{x\in\mathcal{X}} f(x)$ and $y^\star \in \operatorname{argmax}_{y\in\mathcal{Y}} \phi(y)$ satisfies the *minimax principle*: $f(x^\star) = \phi(y^\star) = \psi(x^\star, y^\star)$.

© Yair Carmon, Arun Jambulapati, Liam O'Carroll, and Aaron Sidford;
licensed under Creative Commons License CC-BY 4.0

Convex-concave games are pervasive in algorithm design, machine learning, data analysis, and optimization. For example, the games induced by bilinear objectives, i.e., $\psi(x,y) = x^\top Ay + b^\top x + c^\top y$, where $\mathcal{X}$ and $\mathcal{Y}$ are either the simplex, $\Delta^k \coloneqq \{x \in \mathbb{R}^k_{\geq 0} : \|x\|_1 = 1\}$, or the Euclidean ball, $B^k \coloneqq \{x \in \mathbb{R}^k : \|x\|_2 \leq 1\}$, encompass zero-sum games, linear programming, hard-margin support vector machines (SVMs), and minimum enclosing/maximum inscribed ball [14, 2, 31, 10]. Additionally, the case when $\psi(x,y) = \sum_{i=1}^n y_i f_i(x)$ for some functions $f_i : \mathbb{R}^d \to \mathbb{R}$ and $\mathcal{Y}$ is a subset of the simplex encompasses a variety of distributionally robust optimization (DRO) problems [29, 5] and (for $\mathcal{Y} = \Delta^n$) the problem of minimizing the maximum loss [6, 8, 4].

In this paper, we study the following question:

> *Given only a black-box oracle which solves (regularized versions of)* (P) *to ϵ accuracy, and a black-box oracle for computing an exact dual best response $y_x \coloneqq \operatorname{argmax}_{y\in\mathcal{Y}} \psi(x,y)$ to any primal point $x \in \mathcal{X}$, can we extract an ϵ-optimal solution to* (D)*?*

We develop a general *dual-extraction framework* which answers this question in the affirmative. We show that as long as these oracles can be implemented as cheaply as obtaining an ϵ-optimal point of (P), then our framework can obtain an ϵ-optimal point of (D) at the same cost as that of obtaining an ϵ-optimal point of (P), up to logarithmic factors. We then instantiate our framework to obtain new state-of-the-art results in the settings of bilinear matrix games and DRO. Finally, as evidence of its broader applicability, we show that our framework can be used to recover the optimal complexity for computing a stationary point of a smooth convex function.

In the remainder of the introduction we describe our results in greater detail (Section 1.1), give an overview of the dual extraction framework and its analysis (Section 1.2), discuss related work (Section 1.3), and provide a roadmap for the remainder of the paper (Section 1.4).

1.1 Our results

From primal algorithms to dual optimization

We give a general framework which obtains an ϵ-optimal solution to (D) via a sequence of calls to two black-box oracles: (i) an oracle for obtaining an ϵ-optimal point of a regularized version of (P), and (ii) an oracle for obtaining a dual best response $y_x \coloneqq \operatorname{argmax}_{y\in\mathcal{Y}} \psi(x,y)$ for a given $x \in \mathcal{X}$. In particular, we show it is always possible to obtain an ϵ-optimal point to (D) with at most a logarithmic number of calls to each of these oracles, where the regularized primal optimization oracle is always called to an accuracy of ϵ over a logarithmic factor. We also provide an alternate scheme (or more specifically choice of parameters) for applications where the cost of obtaining an ϵ-optimal point of the regularized primal problem decreases sufficiently as the regularization level increases. In such cases, e.g., in our stationary point application, it is possible to avoid even logarithmic factor increases in computational complexity for approximately solving (D) relative to the complexity of approximately solving (P).

Application 1: Bilinear matrix games

In this application, $\psi(x,y) \coloneqq x^\top Ay$ for a matrix $A \in \mathbb{R}^{d\times n}$, $\mathcal{Y}$ is the simplex Δ^n, and $\mathcal{X}$ is either the simplex Δ^d or the unit Euclidean ball B^d. Recently, [8] gave a new state-of-the-art runtime in certain parameter regimes of $\widetilde{O}(nd + n(d/\epsilon)^{2/3} + d\epsilon^{-2})$ for obtaining an expected ϵ-optimal point for the primal problem (P) for this setup. However, unlike previous algorithms

for bilinear matrix games (see Section 1.3 for details), their algorithm does not return an ϵ-optimal solution for the dual (D), and their runtime is not symmetric in the dimensions n and d. As a result, it was unclear whether the same runtime is achievable for obtaining an ϵ-optimal solution of the dual (D). We resolve this question by applying our general framework to achieve an expected ϵ-optimal point of (D) with runtime $\widetilde{O}(nd + n(d/\epsilon)^{2/3} + d\epsilon^{-2})$. We then observe (see Corollary 22) that in the setting where $\mathcal{X} = \Delta^d$, our result can equivalently be viewed as a new state-of-the-art runtime of $\widetilde{O}(nd + d(n/\epsilon)^{2/3} + n\epsilon^{-2})$ for obtaining an ϵ-optimal point of the primal (P) due to the symmetry of ψ and the constraint sets.

Application 2: CVaR at level α DRO

In this application, $\psi(x, y) \coloneqq \sum_{i=1}^n y_i f_i(x)$ for convex, bounded, and Lipschitz loss functions $f_i : \mathbb{R}^d \to \mathbb{R}$, $\mathcal{X}$ is a convex, compact set, and $\mathcal{Y} \coloneqq \left\{y \in \Delta^n : \|y\|_\infty \le \frac{1}{\alpha n}\right\}$ is the CVaR at level α uncertainty set for $\alpha \in [1/n, 1]$. The primal (P) is a canonical and well-studied DRO problem, and corresponds to the average of the top α fraction of the losses. We consider this problem given access to a first-order oracle that, when queried at $x \in \mathbb{R}^d$ and $i \in [n]$, outputs $(f_i(x), \nabla f_i(x))$. Ignoring dependencies other than α, the target accuracy $\epsilon > 0$, and the number of losses n for brevity, [29] gave a matching upper and lower bound (up to logarithmic factors) of $\widetilde{O}(\alpha^{-1}\epsilon^{-2})$ queries to obtain an expected ϵ-optimal point of the primal (P). However, the best known query complexity for obtaining an expected ϵ-optimal point of the dual (D) was $\widetilde{O}(n\epsilon^{-2})$ prior to this paper (see Section 1.3 for details). Applying our general framework to this setting, we obtain an algorithm with a new state-of-the-art query complexity of $\widetilde{O}(\alpha^{-1}\epsilon^{-2} + n)$ for obtaining an expected ϵ-optimal point of the dual (D). In particular, note that this complexity is nearly linear in n when $\epsilon \ge (\alpha n)^{-2}$.

Application 3: Obtaining stationary points of convex functions

In this application, we show that our framework yields an alternative optimal approach for computing an approximate critical point of a smooth convex function given a gradient oracle. Specifically, for $\gamma > 0$ and convex and β-smooth $h : \mathbb{R}^n \to \mathbb{R}$, in Section 5, we give an algorithm which computes $x \in \mathbb{R}^n$ such that $\|\nabla h(x)\|_2 \le \gamma$ using $O\left(\sqrt{\beta\Delta}/\gamma\right)$ gradient queries, where $\Delta \coloneqq h(x_0) - \inf_{x \in \mathbb{R}^n} h(x)$ is the initial suboptimalityf. While this optimal complexity has been achieved before [24, 37, 15, 28, 27], that we achieve it is a consequence of our general framework illustrates its broad applicability.

For this application, we instantiate our framework with $\psi(x, y) \coloneqq \langle x, y\rangle - h^*(y)$, where $h^* : \mathbb{R}^n \to \mathbb{R}$ denotes the convex conjugate of h. (For reasons discussed in Section 5, we actually first substitute h for an appropriately regularized version of h, call it f, before applying the framework, but the following discussion still holds with respect to f.) This objective function ψ is known as the *Fenchel game* and has been used in the past to recover classic convex optimization algorithms (e.g., the Frank-Wolfe algorithm and Nesterov's accelerated methods) via a minimax framework [1, 43, 12, 23]. In the Fenchel game, a dual best response corresponds to a gradient evaluation:

$$\operatorname*{argmax}_{y \in \mathbb{R}^n} \{\langle x, y\rangle - h^*(y)\} = \nabla h(x),$$

and we show that approximately optimal points for the dual objective (D) must have small norm. As a result, obtaining an approximately optimal dual point y as a best response to a primal point x yields a bound on the norm of $y = \nabla f(x)$. Furthermore, we note that in this setting, adding regularization to ψ with respect to an appropriate choice of

distance-generating function (namely h^*) is equivalent to rescaling and recentering the primal function f, as well as the point at which a gradient is taken in the dual best response computation (cf. Lemma 14 in the full version). Thus, the properties of the Fenchel game extend naturally to appropriately regularized versions of ψ.

1.2 Overview of the framework and analysis

We now give an overview of the dual-extraction framework. Our framework applies generally to a set of assumptions given in Section 3.1 (cf. Definition 9), but for now we specialize to the assumptions given above, namely: (i) the constraint sets $\mathcal{X}$ and $\mathcal{Y}$ are nonempty, compact, and convex; and (ii) ψ is differentiable and convex-concave. Throughout this section, let $\|\cdot\|$ denote any norm on $\mathbb{R}^n$ and assume that the dual function, ϕ, is L-Lipschitz with respect to $\|\cdot\|$.[1] Let $r : \mathbb{R}^n \to \mathbb{R}$ denote a differentiable distance-generating function (dgf) which is μ_r-strongly convex with respect to $\|\cdot\|$ for $\mu_r > 0$,[2] and let $V_u(v) \coloneqq r(v) - r(u) - \langle \nabla r(u), v - u \rangle$ denote the associated Bregman divergence. For the sake of illustration, it may be helpful to consider the choices $\|\cdot\| \coloneqq \|\cdot\|_2$, $r(u) \coloneqq \frac{1}{2}\|u\|_2^2$, $\mu_r = 1$, and $V_u(v) = \frac{1}{2}\|u - v\|_2^2$ in the following, in which case relative strong convexity with respect to r is equivalent to the standard notion of strong convexity with respect to $\|\cdot\|_2$.

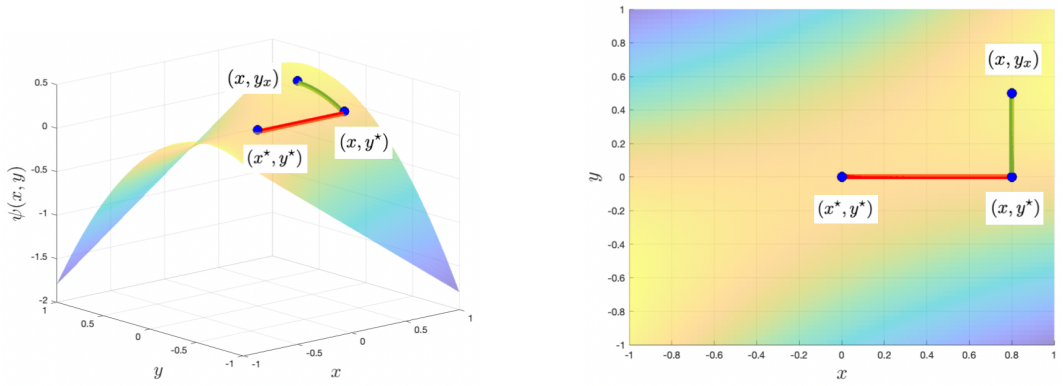
How should we obtain an ϵ-optimal point for (D) using the two oracles discussed previously, namely: (i) an oracle for approximately solving a regularized primal objective, and (ii) an oracle for computing a dual best response? We call (i) a dual-regularized primal optimization (DRPO) oracle and (ii) a dual-regularized best response (DRBR) oracle; their formal definitions are given in Section 3.1. Note that to solve (D), one cannot simply solve the primal problem (P) to high accuracy and then compute a dual best response. Consider $\psi(x, y) = xy$ with $\mathcal{X} = \mathcal{Y} = [-1, 1]$; clearly $x^\star = y^\star = 0$, but for any x arbitrarily close to $x^\star$, the dual best response is either -1 or 1.

The key observation underlying our framework is that if $\psi(x, \cdot)$ is strongly concave for a given $x \in \mathcal{X}$, it is possible to upper bound the distance between the best response $y_x \coloneqq \operatorname{argmax}_{y \in \mathcal{Y}} \psi(x, y)$ and the dual optimum $y^\star$ in terms of the primal suboptimality of x. Figure 1 illustrates why this should be the case when subtracting a quadratic regularizer in y (so that $\psi(x, \cdot)$ is strongly concave) to the preceding example of $\psi(x, y) = xy$. We generalize this intuition in the following lemma (replacing strong concavity with relative strong concavity and a distance bound with a divergence bound), which is itself generalized further and proven in Section 3:

▶ **Lemma 1** (Lemma 3 from the full version specialized). *For a given $x \in \mathcal{X}$, suppose $-\psi(x, \cdot)$ is μ-strongly convex relative to the dgf r for some $\mu > 0$. Then $y_x \coloneqq \operatorname{argmax}_{y \in \mathcal{Y}} \psi(x, y)$ satisfies*

$$V_{y_x}(y^\star) \leq \frac{f(x) - f(x^\star)}{\mu}.$$

[1] This is a weak assumption since we ensure at most a logarithmic dependence on L; see Remark 5.
[2] Section 2 gives the general setup for a distance-generating function which also covers the case where $\operatorname{dom} r \neq \mathbb{R}^n$.



(a) **(b)**

Figure 1 An example to give intuition behind Lemma 1. Here, $\psi(x, y) = xy - 0.8y^2$, $(x^\star, y^\star) = (0, 0)$, $x = 0.8$, and $y_x = 0.5$. To see why it is possible to bound $|y^\star - y_x|$ in terms of the primal suboptimality $f(x) - f(x^\star)$, note that by the strong concavity of $\psi(x, \cdot)$ and the fact that y_x is the maximizer of $\psi(x, \cdot)$ over $\mathcal{Y}$, we can upper bound $|y^\star - y_x|$ in terms of $\psi(x, y_x) - \psi(x, y^\star)$ (the vertical drop over the green line) via a standard strong-concavity inequality. In turn, $\psi(x, y_x) - \psi(x, y^\star)$ can be upper bounded by $\psi(x, y_x) - \psi(x^\star, y^\star) = f(x) - f(x^\star)$ (the vertical drop over the green line plus the vertical drop over the red line) due to the fact that $\psi(x^\star, y^\star) \le \psi(x, y^\star)$ by the optimality of $x^\star$.

A first try

In particular, Lemma 1 suggests the following approach: Define "dual-regularized" versions of ψ, ϕ, f as follows for $\lambda > 0$ and $y_0 \in \mathcal{Y}$:

$$\begin{aligned} \psi_1(x, y) &\coloneqq \psi(x, y) - \lambda V_{y_0}(y), \\ f_1(x) &\coloneqq \max_{y \in \mathcal{Y}} \psi_1(x, y), \\ \phi_1(y) &\coloneqq \min_{x \in \mathcal{X}} \psi_1(x, y)\,. \end{aligned}$$

(Here, the subscript 1 denotes one level of regularization and will be extended later.) For any $x \in \mathcal{X}$, note that $-\psi_1(x, \cdot)$ is λ-strongly convex relative to r, in which case Lemma 1 applied to ψ_1 yields

$$V_{y_x}(y_1^\star) \le \frac{f_1(x) - f_1(x_1^\star)}{\lambda}, \tag{1}$$

for $y_1^\star \coloneqq \operatorname{argmax}_{y \in \mathcal{Y}} \phi_1(y)$, $x_1^\star \in \operatorname{argmin}_{x \in \mathcal{X}} f_1(x)$, and $y_x \coloneqq \operatorname{argmax}_{y \in \mathcal{Y}} \psi_1(x, y)$. Then note

$$\phi(y_1^\star) \ge \phi_1(y_1^\star) \ge \phi_1(y^\star) = \min_{x \in \mathcal{X}} \left\{\psi(x, y^\star) - \lambda V_{y_0}(y^\star)\right\} = \phi(y^\star) - \lambda V_{y_0}(y^\star)\,, \tag{2}$$

where the first inequality follows since $\phi \ge \phi_1$ pointwise. Then by the L-Lipschitzness of ϕ and μ_r-strong convexity of r, it is straightforward to bound the suboptimality of y_x as

$$\phi(y^\star) - \phi(y_x) \le \lambda V_{y_0}(y^\star) + L\sqrt{\frac{2(f_1(x) - f_1(x_1^\star))}{\mu_r \lambda}}. \tag{3}$$

Consequently, an ϵ-optimal point for (D) can be obtained via our oracles as follows: Set $\lambda \leftarrow \frac{\epsilon}{2V_{y_0}(y^\star)}$, and use the DRPO oracle on the regularized primal problem to obtain $x \in \mathcal{X}$ such that

$$f_1(x) - f_1(x_1^\star) \le \frac{\epsilon^3 \mu_r}{16L^2 \cdot V_{y_0}(y^\star)}\,. \tag{4}$$

Then the best response to x with respect to ψ_1, namely $y_x \coloneqq \operatorname{argmax}_{y \in \mathcal{Y}} \psi_1(x, y)$, is ϵ-optimal by (3). However, a typical setting in our applications is $V_{y_0}(y^\star) = \Omega(1)$, $\mu_r = 1$, and $L \geq 1$, in which case ensuring (4) requires solving the regularized primal problem to $O(\epsilon^3)$ error.

Recursive regularization and the dual-extraction framework

To lower the accuracy requirements, we apply dual regularization recursively. A key issue with the preceding argument is that it required a nontrivial bound on $V_{y_0}(y^\star)$. However, it provided us with a nontrivial bound (1) on $V_{y_x}(y_1^\star)$, the "level-one equivalent" of $V_{y_0}(y^\star)$. This suggests solving f_1 to lower accuracy while still obtaining a bound on $V_{y_x}(y_1^\star)$ due to (1), and then adding regularization centered at y_x with a larger value of λ. Indeed, our framework recursively repeats this process until the total regularization is large enough so that (a term similar to) the right-hand side of (3) can be bounded by ϵ, despite never needing to solve a regularized primal problem to high accuracy.

To more precisely describe our approach, let $\psi_0 \coloneqq \psi, f_0 \coloneqq f, \phi_0 \coloneqq \phi$. Over iterations $k = 1, 2, \ldots, K$, our framework implicitly constructs a sequence of convex-concave games $\psi_k : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$, along with corresponding primal and dual functions $f_k : \mathcal{X} \to \mathbb{R}$ and $\phi_k : \mathcal{Y} \to \mathbb{R}$ respectively, as follows:

$$\begin{aligned} \psi_k(x, y) &\coloneqq \psi_{k-1}(x, y) - \lambda_{k-1} V_{y_{k-1}}(y), \\ f_k(x) &\coloneqq \max_{y \in \mathcal{Y}} \psi_k(x, y), \\ \phi_k(y) &\coloneqq \min_{x \in \mathcal{X}} \psi_k(x, y). \end{aligned} \tag{5}$$

Here, $(\lambda_k \in \mathbb{R}_{>0})_{k=0}^{K-1}$ is a dual-regularization schedule given as input to the framework, and $(y_k \in \mathcal{Y})_{i=0}^{K}$ is a sequence of dual-regularization "centers" generated by the algorithm, with y_0 given as input. For $k \in \{0\} \cup [K]$, it will be useful to let $y_k^\star$ denote a maximizer of ϕ_k over $\mathcal{Y}$ and $x_k^\star$ denote a minimizer of f_k over $\mathcal{X}$, with $y_0^\star \coloneqq y^\star$ and $x_0^\star \coloneqq x^\star$ in particular.

Over the K rounds of recursive dual regularization, we aim to balance two goals:

- On the one hand, we want λ_k to increase quickly so that $-\psi_k(x, \cdot)$ is very strongly convex relative to r, thereby allowing us to apply Lemma 1 with a larger strong convexity constant.
- On the other hand, we want to maintain the invariant that, roughly speaking, $y_k^\star$ is always $\epsilon/2$-optimal for the original dual ϕ. Indeed, we were constrained in choosing λ in (2) to be on the order of $\epsilon / V_{y_0}(y^\star)$ to ensure $y_1^\star$ is $\epsilon/2$-optimal for ϕ. A similar "constraint" on the dual-regularization schedule $(\lambda_k)_{k=0}^{K-1}$ appears when (2) is extended to additional levels of regularization. This prevents us from increasing λ_k too quickly.

In all the applications in this paper we choose $\lambda_k \approx 2\lambda_{k-1}$. λ_0 typically must remain on the order of $\epsilon / V_{y_0}(y^\star)$ due to the second point.

Pseudocode of the framework is given in Algorithm 1. Each successive dual-regularization center y_k is computed via the DRBR oracle (Line 5) as a best response to a primal point x_k obtained via the DRPO oracle (Line 4). In Section 3, we generalize Algorithm 1 (cf. Algorithm 2) in several ways: (i) we allow for stochasticity in the DRPO oracle; (ii) we allow for distance-generating functions r such that $\operatorname{dom} r \neq \mathbb{R}^n$; (iii) we give different but equivalent characterizations of x_k and y_k which facilitate the derivation of explicit expressions for the DRPO and DRBR oracles in applications.

Algorithm 1 Dual-extraction framework (Algorithm 2 specialized).

Input: Initial dual-regularization center $y_0 \in \mathcal{Y}$, iteration count $K \in \mathbb{N}$, dual-regularization schedule $(\lambda_k \in \mathbb{R}_{>0})_{k=0}^{K-1}$, primal-accuracy schedule $(\epsilon_k \in \mathbb{R}_{>0})_{k=1}^{K}$, DRPO and DRBR oracles

$\psi_0 \coloneqq \psi$, $f_0 \coloneqq f$, and $\phi_0 \coloneqq \phi$

for $k = 1, 2, \ldots, K$ **do**

  Define ψ_k, f_k, and ϕ_k as in (5)

  Let $x_k \in \mathcal{X}$ be such that $f_k(x_k) - f_k(x_k^\star) \le \epsilon_k$ // Computed via the DRPO oracle

  $y_k = \operatorname{argmax}_{y \in \mathcal{Y}} \psi_k(x_k, y)$ // Computed via the DRBR oracle

return y_K

Analysis of Algorithm 1

Theorem 2 is our main result for Algorithm 1. We then instantiate Theorem 2 with two illustrative choices of parameters in Corollaries 5 and 7, and defer the proofs of the latter to their general versions in Section 3. All of the remarks below (Remarks 3, 5, 7) are stated with reference to the specialized results in this section (Theorem 2 and Corollaries 4, 6 resp.), but extend immediately to the corresponding general versions (Theorem 15 and Corollaries 16, 17 resp.).

▶ **Theorem 2** (Theorem 15 specialized). *Algorithm 1 returns y_K satisfying*

$$V_{y_K}(u) \le \frac{\epsilon_K}{\Lambda_K} \text{ where } \Lambda_k \coloneqq \sum_{j=0}^{k-1} \lambda_j \text{ for } k \in [K] \tag{6}$$

and $u \in \mathcal{Y}$ is a point with dual suboptimality bounded as

$$\phi(y^\star) - \phi(u) \le \lambda_0 V_{y_0}(y^\star) + \sum_{k=1}^{K-1} \frac{\lambda_k}{\Lambda_k} \epsilon_k. \tag{7}$$

If we additionally assume that ϕ is L-Lipschitz with respect to $\|\cdot\|$, we can directly bound the suboptimality of y_K as

$$\phi(y^\star) - \phi(y_K) \le \lambda_0 V_{y_0}(y^\star) + \sum_{k=1}^{K-1} \frac{\lambda_k}{\Lambda_k} \epsilon_k + L\sqrt{\frac{2}{\mu_r} \frac{\epsilon_K}{\Lambda_K}}. \tag{8}$$

Proof. We claim the first half of Theorem 2 holds with $u \leftarrow y_K^\star$. To see this, note that we can bound the suboptimality of $y_K^\star$ as

$$\begin{aligned}
\phi(y_K^\star) \overset{(i)}{\ge} \phi_K(y_K^\star) \ge \phi_K(y_{K-1}^\star) &= \max_{x \in \mathcal{X}} \left\{ \psi_{K-1}(x, y_{K-1}^\star) - \lambda_{K-1} V_{y_{K-1}}\left(y_{K-1}^\star\right) \right\} \\
&= \phi_{K-1}(y_{K-1}^\star) - \lambda_{K-1} V_{y_{K-1}}\left(y_{K-1}^\star\right) \\
&\overset{(ii)}{\ge} \phi_0(y_0^\star) - \lambda_0 V_{y_0}(y_0^\star) - \sum_{k=1}^{K-1} \lambda_k V_{y_k}(y_k^\star) \\
&\overset{(iii)}{\ge} \phi(y^\star) - \lambda_0 V_{y_0}(y^\star) - \sum_{k=1}^{K-1} \frac{\lambda_k}{\Lambda_k} \epsilon_k,
\end{aligned}$$

where (i) follows since $\phi \geq \phi_K$ pointwise, (ii) follows from repeating the argument in the previous lines recursively (starting by lower bounding $\phi_{K-1}(y^\star_{K-1})$, etc.), and (iii) uses Lemma 1 applied to ψ_k, which yields by Lines 4 and 5 in Algorithm 1:

$$V_{y_k}(y^\star_k) \leq \frac{f_k(x_k) - f_k(x^\star_k)}{\Lambda_k} \leq \frac{\epsilon_k}{\Lambda_k},$$

since $\psi_k(x, \cdot) = \psi(x, \cdot) + \sum_{j=0}^{k-1} \lambda_j V_{y_j}(\cdot)$ is Λ_k-strongly concave relative to $-r$. Thus, we have proven Equation 7, and Equation 6 follows again from Lemma 1 applied to ψ_K. Equation 8 then follows since the fact that r is μ_r-strongly convex with respect to $\|\cdot\|$ and Equation 6 imply

$$\|y_K - y^\star_K\| \leq \sqrt{\frac{2}{\mu_r} V_{y_K}(y^\star_K)} \leq \sqrt{\frac{2}{\mu_r} \frac{\epsilon_K}{\Lambda_K}}. \qquad \blacktriangleleft$$

We give a remark regarding how to pick the parameters $(\lambda_k)_{k=0}^{K-1}$ and $(\epsilon_k)_{k=1}^{K}$ when applying Theorem 2:

▶ Remark 3 (Picking the parameters for Theorem 2). Equation 8 can be interpreted as follows: To ensure y_K is ϵ-optimal for ϕ, it suffices to choose the sequences $(\lambda_k)_{k=0}^{K-1}$ and $(\epsilon_k)_{k=1}^{K}$ so that the right side of (8) is at most ϵ. Then the first term, $\lambda_0 V_{y_0}(y^\star)$, constrains λ_0 to be on the order of $\epsilon / V_{y_0}(y^\star)$. Skipping ahead, the third term, $L\sqrt{\frac{2}{\mu_r}\frac{\epsilon_K}{\Lambda_K}}$, is the reason we always choose $\lambda_k \approx 2\lambda_{k-1}$ in our applications, as this ensures Λ_K is large enough to handle this term with K only needing to be logarithmic in the problem parameters. Then the second term, $\sum_{k=1}^{K-1} \frac{\lambda_k}{\Lambda_k}\epsilon_k$, effectively constrains roughly $\sum_{k=1}^{K-1} \epsilon_k \leq \epsilon$, as $\lambda_k / \Lambda_k \approx 1$.

▶ **Corollary 4** (Corollary 16 specialized). *Suppose ϕ is L-Lipschitz with respect to $\|\cdot\|$, and let $B > 0$ be such that $V_{y_0}(y^\star) \leq B$. Then for any $\epsilon > 0$, and $K \geq \max\left\{\log_2 \frac{L^2 B}{\mu_r \epsilon^2}, 1\right\} + 10$, the output of Algorithm 1 with dual-regularization and primal-accuracy schedules of*

$$\lambda_k = 2^k \frac{\epsilon}{4B} \text{ for } k \in \{0\} \cup [K-1] \text{ and } \epsilon_k = \frac{\epsilon}{4K} \text{ for } k \in [K]$$

satisfies $\phi(y^\star) - \phi(y_K) \leq \epsilon$.

▶ Remark 5. Corollary 4 achieves the stated goal of obtaining an ϵ-optimal point for (D) by running for a number of iterations which depends logarithmically on the problem parameters, and solving each dual-regularized primal subproblem to an accuracy of ϵ divided by a logarithmic factor. Note in particular the logarithmic dependence on the dual divergence bound B and dual Lipschitz constant L, meaning these are weak assumptions. Furthermore, it is clear from the proof of Theorem 2 that ϕ only need be L-Lipschitz on a set containing y_K and $y^\star_K$.

▶ **Corollary 6** (Corollary 17 specialized). *Let $B > 0$ be such that $V_{y_0}(y^\star) \leq B$. Then for any $\epsilon > 0$ and $K \in \mathbb{N}$, the output of Algorithm 1 with dual-regularization and primal-accuracy schedules of*

$$\lambda_k = 2^k \frac{\epsilon}{4B} \text{ for } k \in \{0\} \cup [K-1] \text{ and } \epsilon_k = \frac{\epsilon}{8 \cdot 1.5^k} \text{ for } k \in [K]$$

satisfies

$$\|y_K - u\| \leq \frac{1}{1.5^K} \sqrt{\frac{2B}{\mu_r}},$$

where $u \in \mathcal{Y}$ is a point whose suboptimality is at most ϵ, i.e., $\phi(y^\star) - \phi(u) \leq \epsilon$.

▶ Remark 7. Later calls to the DRPO oracle during the run of Algorithm 1 may be cheaper since there will be a significant amount of dual regularization at that point (namely, $\Lambda_k = \sum_{j=0}^{k-1} \lambda_j$ is large). One can sometimes take advantage of this (in particular, if the cost of a DRPO oracle call scales inverse polynomially with the regularization) to design schedules that avoid even the typically additional multiplicative logarithmic cost of Corollary 4 over the cost of a single DRPO oracle call. In such cases, a choice of schedules similar to those of Corollary 6 is often appropriate. With this choice of schedules, later rounds require very high accuracy. However, if one can argue that the increasing dual regularization Λ_k makes the DRPO oracle call cheaper at a faster rate than the decreasing error ϵ_k makes it more expensive (as we do in Section 5), the total cost of applying the framework may collapse geometrically to the cost of a single DRPO oracle call made with target error approximately ϵ.

We purposely state Corollary 6 without the assumption that ϕ is Lipschitz because that is the form we will use in Section 5. However, it is straightforward to reformulate a version of Corollary 6 with the Lipschitz assumption. Here the focus was to illustrate different possible choices of schedules.

1.3 Related work

Black-box reductions

Our main contribution can be viewed as a black-box reduction from (regularized) primal optimization to dual optimization. Similar black-box reductions exist in the optimization literature. For example, [3] develops reductions between various fundamental classes of optimization problems, e.g., strongly convex optimization and smooth optimization. In a similar vein, the line of work [30, 18, 7] reduces convex optimization to approximate proximal point computation (i.e., regularized minimization).

Bilinear matrix games

Consider the bilinear objective $\psi(x, y) = x^\top A y$ where $\mathcal{X}$ and $\mathcal{Y}$ are either the simplex, $\Delta^k := \{x \in \mathbb{R}^k_{\geq 0} : \|x\|_1 = 1\}$, or the Euclidean ball, $B^k := \{x \in \mathbb{R}^k : \|x\|_2 \leq 1\}$. State-of-the-art methods in regard to runtime for obtaining an approximately optimal primal and/or dual solution can be divided into second-order interior point methods [11, 42] and stochastic first-order methods [22, 10, 9, 8]; see Table 2 in [8] for a summary of the best known runtimes as well as other references. Of importance to this paper, all state-of-the-art algorithms other than that of [8] are either (i) primal-dual algorithms which return both an ϵ-optimal primal and dual solution simultaneously, and/or (ii) achieve runtimes which are symmetric in the primal dimension d and dual dimension n, meaning the cost of obtaining an ϵ-optimal dual solution is the same as that of obtaining an ϵ-optimal primal solution. The algorithm of [8], on the other hand, only returns an ϵ-optimal primal point and further has a runtime which is not symmetric in n and d (see the footnote on the first page of that paper). As a result, solving the dual by simply swapping the roles of the primal and dual variables may be more expensive than solving the primal. (In fact, swapping the variables in this way may not even always be possible without further modifications due to restrictions on the constraint sets.)

CVaR at level α distributionally robust optimization (DRO)

The DRO objectives we study are of the form $\psi(x, y) = \sum_{i=1}^n y_i f_i(x)$, where the functions $f_i : \mathbb{R}^d \to \mathbb{R}$ are convex, bounded, and Lipschitz, and $\mathcal{Y}$, known as the uncertainty set, is a subset of the simplex. This objective corresponds to a robust version of the empirical risk

minimization (ERM) objective where instead of taking an average over the losses (namely, y_i is fixed at $1/n$), larger losses may be given more weight. In particular, in this paper we focus on a canonical DRO setting, *CVaR at level α,* where the uncertainty set is given by $\mathcal{Y} \coloneqq \left\{y \in \Delta^n : \|y\|_\infty \leq \frac{1}{\alpha n}\right\}$ for a choice of $\alpha \in [1/n, 1]$. CVaR DRO, along with its generalization f-divergence DRO, has been of significant interest over the past decade; see [29, 5, 13, 32, 16] and the references therein. [29] is the most relevant to this paper – omitting parameters other than α, the number of losses n, and the target accuracy $\epsilon > 0$, they give a matching upper and lower bound (up to logarithmic factors) of $\widetilde{O}(\alpha^{-1}\epsilon^{-2})$ first-order queries of the form $(\nabla f_i(x), f_i(x))$ to obtain an expected ϵ-optimal point of the primal objective. Their upper bound is achieved by a stochastic gradient method where the gradient estimator is based on a multilevel Monte Carlo (MLMC) scheme [19, 20]. However, the best known complexity for obtaining an expected ϵ-optimal point of the dual of CVaR at level α is $O(n\epsilon^{-2})$ via a primal-dual method based on [33]; see also [13, 32] as well as [5, Appendix A.1], the last of which obtains complexity $\widetilde{O}(n\epsilon^{-2})$ in the more general setting of the uncertainty set being an f-divergence ball.

Stationary point computation

For $\gamma > 0$, convex and β-smooth $h : \mathbb{R}^n \to \mathbb{R}$ with global minimum $z^\star$, and initialization point z_0, consider the problem of computing a point z such that $\|\nabla h(z)\|_2 \leq \gamma$. Two worst-case optimal gradient query complexities for this problem exist in the literature: $O\left(\sqrt{\beta(h(z_0) - h(z^\star))}/\gamma\right)$ and $O\left(\sqrt{\beta\|z_0 - z^\star\|_2}/\gamma\right)$. An algorithm (the OGM-G method) which achieves the former complexity was given in [24], and [37] pointed out that any algorithm which achieves the former complexity can achieve the latter complexity. This is obtainable by running N iterations of any optimal gradient method for reducing the function value, followed by N iterations of a method which achieves the former complexity for reducing the gradient magnitude. In what may be of independent interest, we observe in Section 5.1 that a reduction in the opposite direction is also possible. More broadly, algorithms and frameworks for reducing the gradient magnitude of convex functions have been of much recent interest, and further algorithms and related work for this problem include [25, 27, 26, 28, 15, 37, 21], with lower bounds given in [34, 35].

1.4 Paper organization

In Section 2, we go over notation and conventions for the rest of the paper. We give our general dual-extraction framework and its guarantees in Section 3. In Section 4, we apply our framework to bilinear matrix games and the CVaR at level α DRO problem. Finally, in Section 5 we give an optimal algorithm (in terms of query complexity) for computing an approximate stationary point of a convex and β-smooth function.

2 Notation and conventions

We defer standard notation and conventions to the full version, and only include paper-specific notation here.

For $\psi : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$, we use the notation $\psi(\cdot, y) : \mathbb{R}^d \to \mathbb{R}$ for a fixed $y \in \mathbb{R}^n$ to denote the map $x \mapsto \psi(x, y)$ (and define $\psi(x, \cdot)$ analogously). When we say $\psi(\cdot, y)$ satisfies a property, we mean it satisfies that property for any fixed $y \in \mathbb{R}^n$ (and analogously for $\psi(x, \cdot)$). We let $[K] \coloneqq \{1, 2, \ldots, K\}$, $\Delta^n \coloneqq \left\{x \in \mathbb{R}^n_{\geq 0} : \|x\|_1 = 1\right\}$, and $B^n_r(x) \coloneqq \{x \in \mathbb{R}^n : \|x\|_2 \leq r\}$. In the latter two definitions, we may drop the superscript n if it is clear from context, the

argument x if it is 0, and the subscript r if it is 1. For $y \in \mathbb{R}^n$, we may use either the notation y_i or $[y]_i$ to denote its i-th entry. $\mathbf{1}$ denotes the all-ones vector. For a function f which depends on some inputs $x_1, \ldots, x_k \in \mathbb{R}$, we write $f \leq \mathrm{poly}(x_1, \ldots, x_k)$ to denote the fact that f is uniformly bounded above by a polynomial in $x_1, \ldots, x_k$ as $x_1, \ldots, x_k$ vary. We use the notation f^* for the convex or Fenchel conjugate of f. For $S \subseteq \mathbb{R}^n$, we let $\mathbb{I}_S$ denote the infinite indicator of S, namely $\mathbb{I}_S(x) = 0$ if $x \in S$ and $\mathbb{I}_S(x) = \infty$ if $x \notin S$. For a function $f : S \to [-\infty, \infty]$ initially defined on a strict subset $S \subset \mathbb{R}^n$, we may implicitly extend the domain of f to all of $\mathbb{R}^n$ via its indicator as $f + \mathbb{I}_S$ without additional comment. For a function $f : U \to [-\infty, \infty]$ with $S \subseteq U \subseteq \mathbb{R}^n$, we let $f_S \coloneqq f + \mathbb{I}_S$ denote the restriction of f to S. We note that f_S^* denotes the convex conjugate of f_S (and not f^* restricted to S).

Following [38, Sec. 6.4], we encapsulate the setup for a dgf as follows. See the full version for additional discussion of this definition.

▶ **Definition 8** (dgf setup). *We say $(\mathcal{U}, \mathcal{P}, \|\cdot\|, r)$ is a* dgf setup *over $\mathbb{R}^n$ for closed and convex sets $\mathcal{U} \subseteq \mathcal{P} \subseteq \mathbb{R}^n$ with $\mathcal{U} \cap \operatorname{int} \mathcal{P} \neq \emptyset$ if: (i) the distance-generating function (dgf) $r : \mathcal{P} \to \mathbb{R}$ is convex and continuous over $\mathcal{P}$, differentiable on $\operatorname{int} \mathcal{P}$, and μ_r-strongly convex with respect to the chosen norm $\|\cdot\|$ on $\mathcal{U} \cap \operatorname{int} \mathcal{P}$ for some $\mu_r > 0$; and (ii) either $\lim_{u \to \operatorname{bd} \mathcal{P}} \|\nabla r(u)\|_2 = \infty$ or $\mathcal{U} \subseteq \operatorname{int} \mathcal{P}$.*

For a given dgf setup, we define its induced Bregman divergence $V_u^r(v) \coloneqq r(v) - r(u) - \langle \nabla r(u), v - u \rangle$ for $u \in \operatorname{int} \mathcal{P}, v \in \mathcal{P}$, and drop the superscript r when it is clear from context.

3 Dual-extraction framework

In this section, we provide our general dual-extraction framework and its guarantees. In Section 3.1, we give the general setup, oracle definitions, and assumptions with which we apply and analyze the framework. Section 3.2 contains the statement and guarantees of the framework and Section 3.3 in the full version contains the associated proofs.

3.1 Preliminaries

We bundle all of the inputs to our framework into what we call a *dual-extraction setup*, defined below. Recall that when we say $\psi(x, \cdot)$ satisfies a property, we mean it satisfies that property for any fixed $x \in \mathbb{R}^d$ (and analogously for $\psi(\cdot, y)$).

▶ **Definition 9** (Dual-extraction setup). *A* dual-extraction setup *is a tuple $(\psi, \mathcal{X}, \mathcal{Y}, \mathcal{U}, \mathcal{P}, \|\cdot\|, r)$ where:*

1. *$\psi(x, \cdot)$ is differentiable;*
2. *$\psi(\cdot, y)$ and $\psi(x, \cdot)$ are convex and concave respectively;*
3. *$(\mathcal{U}, \mathcal{P}, \|\cdot\|, r)$ is a dgf setup over $\mathbb{R}^n$ per Definition 8;*
4. *the constraint sets $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \subseteq \mathbb{R}^n$ are nonempty, closed, and convex with $\mathcal{Y} \subseteq \mathcal{U}$ and $\mathcal{Y} \cap \operatorname{int} \mathcal{P} \neq \emptyset$;*
5. *$\mathcal{X}$ is bounded or $\psi(\cdot, y)$ is strongly convex;*
6. *$\mathcal{Y}$ is bounded or $\psi(x, \cdot)$ is strongly concave;*
7. *over all $p \in \mathcal{U} \cap \operatorname{int} \mathcal{P}$ and $w \in \partial \mathbb{I}_{\mathcal{U}}(p)$, the map $y \mapsto \langle w, y \rangle$ is constant over $\mathcal{Y}$.*[3]

[3] In all of our applications, this map will in fact be constant over $\mathcal{U}$.

Assumption 1 is only used in the proofs of Lemma 3 in the full version (the general version of Lemma 1 from Section 1.2) and Corollary 13 in the full version (used to show the framework is well-defined when $\operatorname{dom} r \neq \mathbb{R}^n$). Assumptions 2, 5, and 6 ensure that the minimax optimization problem with objective ψ and constraint sets $\mathcal{X}$ and $\mathcal{Y}$ satisfies the *minimax principle*; see below. Regarding Assumptions 3, 4, and 7, the fact that $\mathcal{Y}$ is potentially a strict subset of $\mathcal{U}$ as well as the necessity of the technical assumption 7 is discussed in Remark 4 in the full version. In particular, Assumption 7 is only used to derive an equivalent formulation of the framework to Algorithm 1 which often allows for easier instantiations in applications, but is not strictly necessary to obtain our guarantees.

While our main results are stated in the full generality of Definition 9, in our applications we only particularize to Definition 10 and Definition 11 introduced below.

▶ **Definition 10** (Unbounded setup). *A $(\psi, \mathcal{X}, \mathcal{Y}, r)$-unbounded setup is a $(\psi, \mathcal{X}, \mathcal{Y}, \mathbb{R}^n, \mathbb{R}^n, \|\cdot\|_2, r)$-dual-extraction setup.*

In other words, in an unbounded setup we choose $\mathcal{U} = \mathcal{P} = \mathbb{R}^n$ and the Euclidean norm, in which case the dgf r can be any differentiable and strongly convex function with respect to $\|\cdot\|_2$. Note that Assumption 7 is trivial as $\partial \mathbb{I}_{\mathcal{U}}(p) = \{0\}$ for all $p \in \mathbb{R}^n$.

▶ **Definition 11** (Simplex setup). *A $(\psi, \mathcal{X}, \mathcal{Y})$-simplex setup is a $(\psi, \mathcal{X}, \mathcal{Y}, \Delta^n, \mathbb{R}^n_{\geq 0}, \|\cdot\|_1, r)$-dual-extraction setup where $r(u) \coloneqq \sum_{i=1}^n u_i \ln u_i$ (with $0 \ln 0 \coloneqq 0$).*

In other words, in a simplex setup we choose $\mathcal{U} = \Delta^n$, $\mathcal{P} = \mathbb{R}^n_{\geq 0}$, we are using the ℓ_1-norm, and the dgf is negative entropy when restricted to the simplex. It is a standard result known as Pinsker's inequality that r is 1-strongly convex over $\Delta^n_{>0}$ with respect to $\|\cdot\|_1$, and the associated Bregman divergence is given by the Kullback-Leibler (KL) divergence $V_u(w) = \sum_{i=1}^n w_i \ln \frac{w_i}{u_i}$ for $u \in \Delta^n_{>0}$ and $w \in \Delta^n$. We verify that Assumption 7 holds in Appendix A.1 in the full version.

Notation associated with a setup

Whenever we instantiate a dual-extraction setup (Definition 9), we use the following notation and oracles associated with that setup without additional comment. We define the associated primal $f : \mathcal{X} \to \mathbb{R}$ and dual $\phi : \mathcal{Y} \to \mathbb{R}$ functions, along with their corresponding primal and dual optimization problems, as they were introduced above in (P) and (D). We let $x^\star \in \operatorname{argmin}_{x \in \mathcal{X}} f(x)$ and $y^\star \in \operatorname{argmax}_{y \in \mathcal{Y}} \phi(y)$ denote arbitrary primal and dual optima. To facilitate the discussion of dual-regularized problems, we define $f_{\lambda,q}(x) : \mathcal{X} \to \mathbb{R}$ as follows

$$f_{\lambda,q}(x) \coloneqq \max_{y \in \mathcal{Y}} \{\psi(x, y) - \lambda V_q(y)\} \text{ for } \lambda > 0 \text{ and } q \in \mathcal{U} \cap \operatorname{int} \mathcal{P}\,.$$

The minimax principle

Assumptions 2, 5, and 6 in Definition 9 guarantee $f(x^\star) = \psi(x^\star, y^\star) = \phi(y^\star)$, which we refer to as the *minimax principle*. See, e.g., [39, 41] as well as Propositions 1.2 and 2.4 in [17, Ch. VI].

Oracle definitions

Our framework assumes black-box access to ψ, $\mathcal{X}$, and $\mathcal{Y}$ via a dual-regularized primal optimization (DRPO) oracle and a dual-regularized dual best response (DRBR) oracle defined below. Note that we generalize the setting of Section 1.2 by allowing the DRPO oracle to return an expected ϵ-optimal point; this is used in our applications in Section 4.

▶ **Definition 12** (DRPO oracle). *A* $(q \in \mathcal{U} \cap \operatorname{int} \mathcal{P}, \lambda > 0, \epsilon_{\mathrm{p}} > 0)$*-dual-regularized primal optimization oracle,* $DRPO(q, \lambda, \epsilon_{\mathrm{p}})$*, returns an expected* ϵ_{p}*-minimizer of* $f_{\lambda,q}$*, i.e., a point* $x \in \mathcal{X}$ *such that* $\mathbb{E} f_{\lambda,q}(x) \le \inf_{x' \in \mathcal{X}} f_{\lambda,q}(x') + \epsilon_{\mathrm{p}}$*, where the expectation is over the internal randomness of the oracle.*

▶ **Definition 13** (DRBR oracle). *A* $(q \in \mathcal{U} \cap \operatorname{int} \mathcal{P}, \lambda > 0, x \in \mathcal{X})$*-dual-regularized best response oracle,* $DRBR(q, \lambda, x)$*, returns* $\operatorname{argmax}_{y \in \mathcal{Y}} \left\{ \psi(x, y) - \lambda V_q(y) \right\}$.

We also define a version of the DRPO oracle, called the DRPOSP oracle, which allows for a failure probability. We include this definition here due to its generality and broad applicability, but it is only used in Section 4.1 since the external result we cite to obtain an expected ϵ_{p}-minimizer of $f_{\lambda,q}$ in that application has a failure probability. We also show in Appendix A.4 in the full version how to boost the success probability of a DRPOSP oracle.

▶ **Definition 14** (DRPOSP oracle). *A* $(q \in \mathcal{U} \cap \operatorname{int} \mathcal{P}, \lambda > 0, \epsilon_{\mathrm{p}} > 0, \delta \in [0, 1))$*-dual-regularized primal optimization oracle with success probability,* $DRPOSP(q, \lambda, \epsilon_{\mathrm{p}}, \delta)$*, returns an expected* ϵ_{p}*-minimizer of* $f_{\lambda,q}$ *with success probability at least* $1 - \delta$*, where the expectation and success probability are over the internal randomness of the oracle.*

3.2 The framework and its guarantees

Algorithm 2 Dual-extraction framework.

Input: $(\psi, \mathcal{X}, \mathcal{Y}, \mathcal{U}, \mathcal{P}, \|\cdot\|, r)$-dual extraction setup (Definition 9), initial dual-regularization center $y_0 \in \mathcal{Y} \cap \operatorname{int} \mathcal{P}$, iteration count $K \in \mathbb{N}$, dual-regularization schedule $(\lambda_k \in \mathbb{R}_{>0})_{k=0}^{K-1}$, primal-accuracy schedule $(\epsilon_k \in \mathbb{R}_{>0})_{k=1}^{K}$, DRPO and DRBR oracles (Definitions 12 and 13)

1 **for** $k = 1, 2, \ldots, K$ **do**
2 $\quad \Lambda_k = \sum_{j=0}^{k-1} \lambda_j$
3 $\quad q_k = \operatorname{argmin}_{q \in \mathcal{U}} \frac{1}{\Lambda_k} \sum_{j=0}^{k-1} \lambda_j V_{y_j}(q)$ // Or, $q_k = \nabla r^*_{\mathcal{U}} \left(\frac{1}{\Lambda_k} \sum_{j=0}^{k-1} \lambda_j \nabla r(y_j) \right)$; see Appendix A.2 in the full version
4 $\quad x_k = \mathrm{DRPO}(q_k, \Lambda_k, \epsilon_k)$ // $\mathbb{E}[f_{\Lambda_k, q_k}(x_k) \mid x_{k-1}] \le \inf_{x \in \mathcal{X}} f_{\Lambda_k, q_k}(x) + \epsilon_k$
5 $\quad y_k = \mathrm{DRBR}(q_k, \Lambda_k, x_k)$ // $y_k = \operatorname*{argmax}_{y \in \mathcal{Y}} \left\{ \psi(x_k, y) - \Lambda_k V_{q_k}(y) \right\}$
6 **return** y_K

We now state the general dual-extraction framework, Algorithm 2, and its guarantees, with proofs in the next section. As mentioned in Section 1.2, Algorithm 2 generalizes Algorithm 1 in three major ways: (i) we allow for stochasticity in the DRPO oracle; (ii) we allow for distance-generating functions r where $\operatorname{dom} r \neq \mathbb{R}^n$; and (iii) we give different but equivalent characterizations of x_k and y_k which often allow for easier instantiations of the framework.

Regarding (iii), consider the case where the DRPO oracle is deterministic and $\operatorname{dom} r = \mathbb{R}^n$ for the sake of discussion. Note that in this case, the definitions of x_k and y_k in Lines 4 and 5 of Algorithm 2 may seem different than those in Lines 4 and 5 of Algorithm 1 at first glance. In particular, x_k in Line 4 of Algorithm 2 is an ϵ_k-minimizer of $x \mapsto \max_{y \in \mathcal{Y}} \{\psi(x, y) - \Lambda_k V_{q_k}(y)\}$ over $\mathcal{X}$, whereas x_k in Line 4 of Algorithm 1 is an ϵ_k-minimizer of $x \mapsto \max_{y \in \mathcal{Y}} \{\psi(x, y) - \sum_{j=0}^{k-1} \lambda_j V_{y_j}(y)\}$ over $\mathcal{X}$. Similarly, $y_k = \operatorname{argmax}_{y \in \mathcal{Y}} \{\psi(x_k, y) - \Lambda_k V_{q_k}(y)\}$ in Line 5 of Algorithm 2, whereas $y_k = \operatorname{argmax}_{y \in \mathcal{Y}} \{\psi(x, y) - \sum_{j=0}^{k-1} \lambda_j V_{y_j}(y)\}$ in Line 5 of Algorithm 1.

In fact, we show in Section 3.3 in the full version that these are equivalent; see Lemma 2 and Remark 4 in the full version. The potential advantage of the expressions in Algorithm 2 compared to those in Algorithm 1 is that they involve only a single regularization term.

Note also that Line 3 of Algorithm 2 gives two equivalent expressions for the iterate q_k; their equivalence is proven in Appendix A.2 in the full version. Also, note that Line 4 is the only potential source of randomness in Algorithm 2; in particular, y_k and q_{k+1} are deterministic upon conditioning on x_k. Finally, we show that Algorithm 2 is well-defined in Appendix A.3 in the full version; in particular, whenever a Bregman divergence $V_u(w)$ is written in Algorithm 2, it is the case that $u \in \mathcal{U} \cap \operatorname{int} \mathcal{P}$. For example, in the context of a simplex setup per Definition 11, this corresponds to $u \in \Delta^n_{>0}$.

We now give the main guarantee for Algorithm 2. See Remark 3 for additional explanation.

▶ **Theorem 15** (Algorithm 2 guarantee)**.** *With K calls to a DRPO oracle and K calls to a DRBR oracle, Algorithm 2 returns y_K satisfying*

$$\mathbb{E}V_{y_K}(u) \le \frac{\epsilon_K}{\Lambda_K},$$

where $u \in \mathcal{Y}$ is a point with expected suboptimality bounded as

$$\phi(y^\star) - \mathbb{E}\phi(u) \le \lambda_0 V_{y_0}(y^\star) + \sum_{k=1}^{K-1} \frac{\lambda_k}{\Lambda_k}\epsilon_k.$$

If we additionally assume that ϕ is L-Lipschitz with respect to $\|\cdot\|$, the expected suboptimality of y_K can be directly bounded as

$$\phi(y^\star) - \mathbb{E}\phi(y_K) \le \lambda_0 V_{y_0}(y^\star) + \sum_{k=1}^{K-1} \frac{\lambda_k}{\Lambda_k}\epsilon_k + L\sqrt{\frac{2}{\mu_r}\frac{\epsilon_K}{\Lambda_K}}. \tag{9}$$

We now particularize Theorem 15 using two exemplary choices of the dual-regularization and primal-accuracy schedules. See Remarks 5 and 7 for additional comments.

▶ **Corollary 16.** *Suppose ϕ is L-Lipschitz with respect to $\|\cdot\|$, and let $B > 0$ be such that $V_{y_0}(y^\star) \le B$. Then for any $\epsilon > 0$, and $K \ge \max\left\{\log_2 \frac{L^2 B}{\mu_r \epsilon^2}, 1\right\} + 10$, the output of Algorithm 2 with dual-regularization and primal-accuracy schedules given by*

$$\lambda_k = 2^k \frac{\epsilon}{4B} \text{ for } k \in \{0\} \cup [K-1] \text{ and } \epsilon_k = \frac{\epsilon}{4K} \text{ for } k \in [K]$$

satisfies $\phi(y^\star) - \mathbb{E}\phi(y_K) \le \epsilon$.

▶ **Corollary 17.** *Let $B > 0$ be such that $V_{y_0}(y^\star) \le B$. Then for any $\epsilon > 0$ and $K \in \mathbb{N}$, the output of Algorithm 2 with dual-regularization and primal-accuracy schedules given by*

$$\lambda_k = 2^k \frac{\epsilon}{4B} \text{ for } k \in \{0\} \cup [K-1] \text{ and } \epsilon_k = \frac{\epsilon}{8 \cdot 1.5^k} \text{ for } k \in [K] \tag{10}$$

satisfies

$$\mathbb{E}\|y_K - u\| \le \frac{1}{1.5^K}\sqrt{\frac{2B}{\mu_r}},$$

where $u \in \mathcal{Y}$ is a point whose expected suboptimality is at most ϵ, i.e., $\phi(y^\star) - \mathbb{E}\phi(u) \le \epsilon$.

4 Efficient maximin algorithms

In this section, we obtain new state-of-the-art runtimes for solving bilinear matrix games in certain parameter regimes (Section 4.1), as well as an improved query complexity for solving the dual of the CVaR at level α distributionally robust optimization (DRO) problem (Section 4.2). In each application, we apply Corollary 16 to compute an ϵ-optimal point for the dual problem at approximately the same cost as computing an ϵ-optimal point for the primal problem (up to logarithmic factors and the cost of representing a dual vector when it comes to CVaR at level α).

4.1 Bilinear matrix games

In this section, we instantiate $\psi(x, y) \coloneqq x^\top A y$ for a matrix $A \in \mathbb{R}^{d\times n}$. Given $p, q \geq 1$, we write $\|A\|_{p\to q} \coloneqq \max_{v\in\mathbb{R}^d, v\neq 0} \frac{\|Av\|_q}{\|v\|_p}$, and use the notation A_{ij}, $A_{i:}$, and $A_{:j}$ for the (i, j) entry, i-th row as a row vector, and j-th column as a column vector. We consider two setups:

▶ **Definition 18** (Matrix games ball setup)**.** *In the* matrix games ball setup, *we set* $\mathcal{X} \coloneqq B^d$ *(the unit Euclidean ball in* $\mathbb{R}^d$*),* $\mathcal{Y} \coloneqq \Delta^n$*, and fix a* $(\psi, \mathcal{X}, \mathcal{Y})$*-simplex setup (Definition 11). We assume* $\|A^\top\|_{2\to\infty} = \max_{i\in[n]} \|A_{:i}\|_2 \leq 1$.

▶ **Definition 19** (Matrix games simplex setup)**.** *In the* matrix games simplex setup, *we set* $\mathcal{X} \coloneqq \Delta^d$, $\mathcal{Y} \coloneqq \Delta^n$, *and fix a* $(\psi, \mathcal{X}, \mathcal{Y})$*-simplex setup (Definition 11). We assume* $\|A^\top\|_{1\to\infty} = \max_{i,j} |A_{ij}| \leq 1$.

Throughout Section 4.1, any theorem, statement, or equation which does not make reference to a specific choice of Definition 18 or 19 applies to both setups. Specializing the primal (P) and dual (D) to this application gives

$$\underset{x\in\mathcal{X}}{\text{minimize}}\, f(x) \text{ for } f(x) \coloneqq \max_{y\in\Delta^n} x^\top A y, \text{ and} \tag{P-MG}$$

$$\underset{y\in\Delta^n}{\text{maximize}}\, \phi(y) \text{ for } \phi(y) \coloneqq \min_{x\in\mathcal{X}} x^\top A y. \tag{D-MG}$$

Regarding the assumptions on the norm of the matrix A in Definitions 18 and 19, note that we can equivalently write $f(x) = \max_{y\in\Delta^n} \sum_{i=1}^n y_i f_i(x)$ with $f_i(x) \coloneqq [A^\top x]_i$. Then the assumptions on the norm of A correspond to ensuring f_i is 1-Lipschitz with respect to the ℓ_2-norm in Definition 18 and ℓ_1-norm in Definition 19 (which in turn implies f is 1-Lipschitz in the respective norms). This normalization is performed to simplify expressions as in [8]. (In particular, [8] also considers the more general problem where each f_i can be any smooth, Lipschitz, convex function.)

Recently, [8, Cor. 8.2] achieved a state-of-the-art runtime in certain parameter regimes of $\widetilde{O}(nd + n(d/\epsilon)^{2/3} + d\epsilon^{-2})$ for obtaining an ϵ-optimal point for (P-MG). However, unlike previous algorithms for (P-MG) (see Section 1.3 for an extended discussion), their algorithm does not yield an ϵ-optimal point for (D-MG) with the same runtime.

Our instantiation of the dual-extraction framework in Algorithm 3 and the accompanying guarantee Theorem 21 resolves this asymmetry between the complexity of obtaining a primal versus dual ϵ-optimal point by obtaining an ϵ-optimal point of (D-MG) with the same runtime of $\widetilde{O}(nd + n(d/\epsilon)^{2/3} + d\epsilon^{-2})$. At the end of Section 4.1, we observe that Theorem 21 also yields a new state-of-the-art runtime for the primal (P-MG) in the setting of Definition 19 due to the symmetry of the constraint sets and ψ.

Algorithm 3 Dual extraction for matrix games.

Input: $(\psi, \mathcal{X}, \Delta^n)$-simplex setup (Definition 11), iteration count $K \in \mathbb{N}$, dual-regularization schedule $(\lambda_k \in \mathbb{R}_{>0})_{k=0}^{K-1}$, primal-accuracy schedule $(\epsilon_k \in \mathbb{R}_{>0})_{k=1}^{K}$, DRPOSP oracle (Definition 14)

$y_0 \coloneqq \frac{1}{n}\mathbf{1}$

for $k = 1, 2, \ldots, K$ **do**

  $\Lambda_k = \sum_{j=0}^{k-1} \lambda_j$

  $[q_k]_i \propto \prod_{j=0}^{k-1} [y_j]_i^{\lambda_j/\Lambda_k},\ \forall i \in [n]$ // Note: $q_k \in \Delta^n$

  $x_k = \mathrm{DRPOSP}(q_k, \Lambda_k, \epsilon_k, \frac{1}{10K})$

  $[y_k]_i \propto [q_k]_i \cdot \exp(\Lambda_k^{-1} \cdot [A^\top x_k]_i),\ \forall i \in [n]$ // $y_k = \operatorname{argmax}_{y \in \Delta^n} \left\{ x_k^\top A y - \Lambda_k V_{q_k}(y) \right\}$

return y_K

Before giving the guarantee Theorem 21 for Algorithm 3, the following lemma provides a runtime bound for the DRPOSP oracle when the success probability is 9/10 (see Appendix B.1 in the full version for the proof). In particular, Lemma 20 shows that adding dual regularization to (P-MG) does not increase the complexity of obtaining an ϵ-optimal point over the guarantee of [8, Cor. 8.2] discussed above.

▶ **Lemma 20** (DRPOSP oracle for matrix games). *In the settings of Definitions 18 and 19, for any $q \in \Delta^n_{>0}$, $\epsilon_{\mathrm{p}} > 0$, and $\lambda > 0$, with success probability at least 9/10, there exists an algorithm which returns an expected ϵ_{p}-optimal point of $f_{\lambda,q}$ with runtime $\widetilde{O}(nd + n(d/\epsilon_{\mathrm{p}})^{2/3} + d\epsilon_{\mathrm{p}}^{-2})$. (Equivalently, per Definition 14, we have that DRPOSP$(q, \lambda, \epsilon_{\mathrm{p}}, 1/10)$ can be implemented with this runtime.)*

Now for the main guarantee (we defer the proof to the full version):

▶ **Theorem 21** (Guarantee for Algorithm 3). *In the settings of Definitions 18 and 19, given target error $\epsilon > 0$ and with success probability at least 9/10, Algorithm 3 with dual-regularization and primal-accuracy schedules given by*

$$\lambda_k = 2^k \frac{\epsilon}{4 \ln n} \text{ for } k \in \{0\} \cup [K-1] \text{ and } \epsilon_k = \frac{\epsilon}{4K} \text{ for } k \in [K]$$

for $K = \left\lceil \max\left\{ \log_2 \frac{\ln n}{\epsilon^2}, 1 \right\} \right\rceil + 10$ returns an expected ϵ-optimal point for (D-MG)*, and can be implemented with runtime $\widetilde{O}(nd + n(d/\epsilon)^{2/3} + d\epsilon^{-2})$.*

The primal perspective

As alluded to above, the guarantee of Theorem 21 also implies a new state-of-the-art runtime for the primal (P-MG) in the setting of Definition 19. This follows because in the matrix games simplex setup, (P-MG) and (D-MG) are symmetric in terms of their constraint sets, so we can obtain an expected ϵ-optimal point for (P-MG) via Theorem 21 by negating and treating (P-MG) as if it were the dual problem. Formally (we defer the proof to the full version):

▶ **Corollary 22** (Guarantee for (P-MG) in the matrix games simplex setup). *In the setting of Definition 19, there exists an algorithm which, given target error $\epsilon > 0$ and with success probability at least 9/10, returns an expected ϵ-optimal point for* (P-MG) *with runtime $\widetilde{O}(nd + d(n/\epsilon)^{2/3} + n\epsilon^{-2})$.*

See the full version for a discussion of how this runtime compares to the prior art.

4.2 CVaR at level α DRO

In this section, we instantiate $\psi(x, y) \coloneqq \sum_{i=1}^n y_i f_i(x)$ for convex, bounded, and G-Lipschitz (with respect to the Euclidean norm) functions $f_i : \mathbb{R}^d \to \mathbb{R}$.[4] Given a compact, convex set $\mathcal{X}$ and $\alpha \in [1/n, 1]$, the primal and dual problem for CVaR at level α are as follows (we explain the reason for the notation $\bar{f}$ as opposed to f in the full version; in short, we apply the framework to a proxy objective):

$$\underset{x\in\mathcal{X}}{\text{minimize}}\ \bar{f}(x) \qquad \text{for } \bar{f}(x) \coloneqq \max_{y\in\Delta^n, \|y\|_\infty \le \frac{1}{\alpha n}} \sum_{i=1}^n y_i f_i(x), \text{ and} \tag{P-CVaR}$$

$$\underset{y\in\Delta^n, \|y\|_\infty \le \frac{1}{\alpha n}}{\text{maximize}}\ \phi(y) \text{ for } \phi(y) \coloneqq \min_{x\in\mathcal{X}} \sum_{i=1}^n y_i f_i(x). \tag{D-CVaR}$$

Our complexity model in this section is the number of computations of the form $(f_i(x), \nabla f_i(x))$ for $x \in \mathcal{X}$ and $i \in [n]$. We refer to the evaluation of $(f_i(x), \nabla f_i(x))$ for a given $x \in \mathcal{X}$ and $i \in [n]$ as a single *first-order query*. Omitting the Lipschitz constant G and bounds on the range of the f_i's and size of $\mathcal{X}$ for clarity, [29, Sec. 4] gave an algorithm which returns an expected[5] ϵ-optimal point of (P-CVaR) with $\widetilde{O}(\alpha^{-1}\epsilon^{-2})$ first-order queries, and also proved a matching lower bound up to logarithmic factors when n is sufficiently large. However, to the best of our knowledge, the best known complexity for obtaining an expected ϵ-optimal point of (D-CVaR) is $\widetilde{O}(n\epsilon^{-2})$ via a primal-dual method based on [33]; see also [13, 32, 5]. In our main guarantee for this section, Theorem 24, we apply Algorithm 2 to obtain an expected ϵ-optimal point of (D-CVaR) with complexity $\widetilde{O}(\alpha^{-1}\epsilon^{-2} + n)$, which always improves upon or matches $\widetilde{O}(n\epsilon^{-2})$ since $\alpha \in [1/n, 1]$.

Toward stating our main guarantee, we encapsulate the formal assumptions of [29, Sec. 2] in the following definition:

▶ **Definition 23** (CVaR at level α setup)**.** *We assume $\mathcal{X}$ is nonempty, closed, convex, and satisfies $\|x - y\|_2 \le R$ for all $x, y \in \mathcal{X}$. We also assume, for all $i \in [n]$, that f_i is convex, G-Lipschitz with respect to $\|\cdot\|_2$, and satisfies $f_i(x) \in [0, M]$ for all $x \in \mathcal{X}$.*

We ultimately obtain the following guarantee via Algorithm 2. Note that the upper bound on ϵ in Theorem 24 is without loss of generality since if $\epsilon \ge M$, any feasible point is ϵ-optimal. We defer the proof to the full version.

▶ **Theorem 24** (Guarantee for (D-CVaR))**.** *In the setting of Definition 23 with target error $\epsilon \in (0, 4M)$ and $\alpha \in [1/n, 1]$, there exists an algorithm which computes an expected ϵ-optimal point of* (D-CVaR) *with complexity $\widetilde{O}(n + G^2R^2\alpha^{-1}\epsilon^{-2})$.*

5 Obtaining critical points of convex functions

In this section, our goal is to obtain an approximate critical point of a convex, β-smooth function $h : \mathbb{R}^n \to \mathbb{R}$, given access to a gradient oracle for h. We show that our general framework yields an algorithm with the optimal query complexity for this problem. In

[4] Note that we do not require the functions f_i to be differentiable. Here, it is important that Definition 9 only requires $\psi(x, \cdot)$ to be differentiable.

[5] To be precise, [29] gives a $\widetilde{O}(\alpha^{-1}\epsilon^{-2})$-complexity high probability bound in Theorem 2. They do not state a $\widetilde{O}(\alpha^{-1}\epsilon^{-2})$-complexity expected suboptimality bound explicitly in a theorem, but they note in the text above Theorem 2 that such a bound follows immediately from Propositions 3 and 4 in their paper.

Section 5.1, we give the formal problem definition and some important preliminaries. In Section 5.2, we give the setup for applying our main framework Algorithm 2 to this problem and a sketch of why the resulting algorithm works. In Section 5.3, we formally state the resulting algorithm for obtaining an approximate critical point of h and prove that it achieves the optimal rate using the guarantees associated with Algorithm 2.

5.1 Preliminaries for Section 5

Throughout Section 5, we fix $\|\cdot\|$ to be the standard Euclidean norm over $\mathbb{R}^n$. We assume $h : \mathbb{R}^n \to \mathbb{R}$ is convex, β-smooth with respect to $\|\cdot\|$, and $\Delta \coloneqq h(x_0) - \inf_{x\in\mathbb{R}^n} h(x) < \infty$ for an arbitrary initialization point $x_0 \in \mathbb{R}^n$. We access h through a gradient oracle. For $\gamma > 0$, our goal will be to obtain a γ-critical point of h, i.e., a point $x \in \mathbb{R}^n$ such that $\|\nabla h(x)\| \le \gamma$. Instead of operating on h itself, our algorithm will operate on a regularized version of h:

$$f(x) \coloneqq h(x) + \frac{\gamma^2}{16\Delta}\|x - x_0\|^2. \tag{11}$$

This notation was chosen to mirror the notation of Section 3.1; f will be the primal function when we apply the framework. Let $x_f^\star$ denote the unique global minimum of f. The following corollary of Lemma 13 in Appendix C in the full version summarizes the key properties of f:

▶ **Corollary 25** (Properties of the regularized function f)**.** *We have*
1. $\|x_f^\star - x_0\| \le 4\Delta/\gamma$.
2. *If* $u \in \mathbb{R}^n$ *is such that* $\|\nabla f(u)\| \le \gamma/4$*, then* $\|\nabla h(u)\| \le \gamma$.

Proof. This follows immediately from Lemma 13 in the full version with $\alpha \leftarrow \frac{\gamma^2}{8\Delta}$ and $\nu \leftarrow \gamma/4$. ◀

The second part of Corollary 25 says that to find a γ-critical point of h, it suffices to find a $(\gamma/4)$-critical point of f. Furthermore, clearly a single query to ∇h suffices to obtain ∇f at a point. As a result, we will focus on finding a $(\gamma/4)$-critical point of f. Furthermore, Corollary 25 may be of independent interest since it trivially allows one to achieve a gradient query complexity of $O\left(\sqrt{\beta\Delta}/\gamma\right)$ via a method which achieves query complexity $O\left(\sqrt{\beta\|x_0 - x_h^\star\|/\gamma}\right)$ (for $x_h^\star$ defined as some minimizer of h over $\mathbb{R}^n$, assuming one exists); see Section 1.3.

The reason we perform this regularization before applying our framework is it enables us to obtain a sufficiently tight bound on $V_{y_0}(y^\star)$ (equivalently, a small enough value of B when we ultimately apply Corollary 17). It is possible to apply the framework more directly to h, but it is not clear how to do so in a way that achieves an optimal complexity.

Finally, we provide a notation guide for Section 5 in Table 1, which may be useful to reference as additional notation is introduced in Sections 5.2 and 5.3.

5.2 Instantiating the framework

For this application, we instantiate

$$\psi(x, y) \coloneqq \langle x, y\rangle - f^*(y).$$

Recall that ψ is the *Fenchel game* [1, 43, 12, 23]; see Section 1.1 for a discussion of why it is a natural choice in this setting. For the rest of Section 5, we fix a $(\psi, \mathcal{X} \coloneqq B_R^n(x_0), \mathcal{Y} \coloneqq \mathbb{R}^n, f^*)$-unbounded setup (Definition 10) with $R \coloneqq 5\Delta/\gamma$. f^* is a valid choice for the dgf because

Table 1 Notation guide for Section 5.

Notation	Description	Section
$\|\cdot\|$	Euclidean norm	5.1
h	Convex, β-smooth function	
γ	Target critical point error for h	
x_0	Arbitrary initialization point	
Δ	$h(x_0) - \inf_{x\in\mathbb{R}^n} h(x) < \infty$	
$f(x)$	$h(x) + \frac{\gamma^2}{16\Delta}\|x - x_0\|^2$	
$x_f^\star$	The global minimizer of f	
$\psi(x, y)$	$\langle x, y\rangle - f^*(y)$	5.2
R	$5\Delta/\gamma$	
$\mathcal{X}$	$B_R^n(x_0)$	
$\mathcal{Y}$	$\mathbb{R}^n$	
dgf r	f^*	
$\phi(y)$	$\langle x_0, y\rangle - R\|y\| - f^*(y)$	
λ_k	$2^k/32$	5.3
ϵ_k	$\Delta/(64 \cdot 1.5^k)$	
CGM	Fast composite gradient method oracle	

f^* is differentiable and $\left(\beta + \frac{\gamma^2}{8\Delta}\right)^{-1}$-strongly convex [38, Thm. 6.11]. The strong convexity of f^* also implies that Assumption 6 holds. Note that the associated primal function $x \mapsto \max_{y\in\mathbb{R}^n} \psi(x, y)$ is precisely $f^{**} = f$ (hence the choice of notation in (11)), and the dual function is given by

$$\phi(y) = \min_{x\in B_R^n(x_0)} \{\langle x, y\rangle - f^*(y)\} = \left\langle x_0 - R\frac{y}{\|y\|}, y\right\rangle - f^*(y) = \langle x_0, y\rangle - R\|y\| - f^*(y).$$

Next, the following lemma fulfills part of the outline given in Section 1.1 by showing that approximately optimal points for the dual objective (D) must have small norm. We defer the proof to the full version.

▶ **Lemma 26** (Bounding the norm by dual suboptimality). *If $y \in \mathbb{R}^n$ is ϵ-optimal for* (D) *for some $\epsilon > 0$, then $\|y\| \le \epsilon\gamma/\Delta$.*

We now derive the oracles of Definitions 12 and 13. Regarding Definition 12, for the rest of Section 5 we restrict DRPO$(\cdot)$ to denote a deterministic implementation of the DRPO oracle, since we can always obtain a deterministic implementation in this application. Then the following corollary is an immediate consequence of a more general lemma given in Appendix C in the full version which characterizes the properties of the Fenchel game with added dual regularization; see also Section 1.1.

▶ **Corollary 27.** *The set of valid output points of $DRPO(q \in \mathbb{R}^n, \lambda > 0, \epsilon_{\mathrm{p}} > 0)$ is precisely*

$$\operatorname*{argmin}^{\epsilon_{\mathrm{p}}}_{x\in B_R^n(x_0)} (1+\lambda)\cdot f\left(\frac{x + \lambda\nabla f^*(q)}{1+\lambda}\right), \text{ and}$$

$$DRBR(q \in \mathbb{R}^n, \lambda > 0, x \in B_R^n(x_0)) = \nabla f\left(\frac{x + \lambda\nabla f^*(q)}{1+\lambda}\right).$$

Proof. Apply Lemma 14 in the full version with $g \leftarrow f$. ◀

Taken together, Lemma 26 and Corollary 27 nearly immediately imply that Algorithm 2 can be applied to the above setup to obtain a $(\gamma/4)$-critical point of f (and therefore a γ-critical point of h). In particular, we will apply the schedules of Corollary 17 to certify that the output y_K of Algorithm 2 is close in distance to an ϵ-optimal point for (D) for an appropriate choice of $\epsilon > 0$. Then Lemma 26 and a triangle inequality yield a bound on $\|y_K\|$. Finally, since

$$y_K \coloneqq \mathrm{DRBR}(q_K, \Lambda_K, x_K) = \nabla f\left(\frac{x_K + \Lambda_K \nabla f^*(q_K)}{1 + \Lambda_K}\right)$$

by Corollary 27, we have that $\frac{x_K+\Lambda_K\nabla f^*(q_K)}{1+\Lambda_K}$ is an approximate critical point of f (and therefore h). One may worry about the presence of $\nabla f^*(q_K)$ here and, more generally, the presence of $\nabla f^*(q)$ in the expressions for the oracles in Corollary 27. However, $\nabla f^*(\cdot)$ never needs to be evaluated explicitly since per the alternate expression for q_k given in Line 3 of Algorithm 2, note that q_k was itself computed as the gradient of f at a point (recall the dgf is f^* and $f = f^{**}$), in which case ∇f^* simply undoes this operation by Lemma 16 in the full version.

We formalize this sketch and provide a complexity guarantee in the next section. We also reframe this sketch and treat the sequence of $\frac{x_k+\Lambda_k\nabla f^*(q_k)}{1+\Lambda_k}$ terms as our iterates (as opposed to the sequence of x_k's), as this leads to a simpler statement and interpretation of the resulting algorithm.

5.3 The resulting algorithm and guarantee

We now formalize the sketch given at the end of the previous section, state the resulting algorithm, and provide a complexity guarantee. But first, we define a subroutine which will be used by the algorithm to implement the DRPO oracle:

▶ **Definition 28** (CGM oracle [40, 36]). *A $(\zeta > 0, w \in \mathbb{R}^n, \epsilon > 0)$-fast composite gradient method oracle, $CGM(\zeta, w, \epsilon)$, returns an ϵ-minimizer of f over $x \in B_\zeta^n(w)$, i.e., an element of $\operatorname{argmax}^{\epsilon}_{x \in B_\zeta^n(w)} f(x)$, using at most $O\left(1 + \sqrt{\frac{\beta\zeta^2}{\epsilon}}\right)$ queries to ∇f.*

For example, implementations with a small constant can be found in [40] or [36, Sec. 6.1.3]. The implementation of the CGM oracle falls under fast gradient methods for composite minimization, where letting g denote a convex, β-smooth function and Ψ a "simple regularizer" (a quadratic in our case), the goal is to minimize $\tilde{g}(x) \coloneqq g(x) + \Psi(x)$ with the same complexity as it takes to minimize g. The domain constraint can also be baked into the regularizer Ψ by adding an indicator.

Our method for computing a γ-critical point of h is given in Algorithm 4, with the associated guarantee in Theorem 30. We note that the decision to introduce the equivalent notation z_0 for x_0 in Line 1 is aesthetic (to make Line 5 simpler to state and interpret). Furthermore, we state Algorithm 4 for general schedules $(\lambda_k)_{k=0}^{K-1}$ and $(\epsilon_k)_{k=1}^{K}$ for clarity, but ultimately we will choose the schedules given in Theorem 30, which correspond to particularizing the schedules of Corollary 17 to this setting. With this choice of schedules, $\Lambda_k \approx 2^k$ and $\epsilon_k \approx \Delta/1.5^k$ so that $\frac{\epsilon_k}{1+\Lambda_k} \approx \Delta/3^k$. As a result, Algorithm 4 can be interpreted as optimizing f in a sequence of balls where the radius and target error are both decreasing geometrically, and the center is a convex combination of the past iterates. While we choose the iteration count K to be logarithmic in the problem parameters, we avoid multiplicative

logarithmic factors in the total complexity because the ratio ζ^2/ϵ in the complexity of the CGM oracle call (to borrow the notation of Definition 28) in Line 5 of Algorithm 4 is $\approx \frac{R^2}{4^k} \cdot \frac{3^k}{\Delta}$ at the k-th iteration, meaning it is collapsing geometrically.

Algorithm 4 Algorithm for obtaining a γ-critical point of h.

Input: Sequences $(\lambda_k)_{k=0}^{K-1}$ and $(\epsilon_k)_{k=1}^{K}$ specified in Theorem 30, iteration count $K \in \mathbb{N}$, CGM oracle (Definition 28)

1 $z_0 := x_0$
2 **for** $k = 1, 2, \ldots, K$ **do**
3 $\quad \Lambda_k = \sum_{j=0}^{k-1} \lambda_j$
4 $\quad w_k = \frac{z_0 + \sum_{j=0}^{k-1} \lambda_j z_j}{1+\Lambda_k}$
5 $\quad z_k = \mathrm{CGM}\left(\frac{R}{1+\Lambda_k}, w_k, \frac{\epsilon_k}{1+\Lambda_k}\right)$ // $z_k \in \operatorname{argmin}^{\epsilon_k/(1+\Lambda_k)}_{z \in B^n_{R/(1+\Lambda_k)}(w_k)} f(z)$
6 **return** z_K

Toward analyzing Algorithm 4, we first connect the sequence of iterates z_k produced by Algorithm 4 to the sequence of iterates x_k, y_k, q_k produced by Algorithm 2 with the same input parameters. Namely, we are formalizing the comment made at the end of Section 5.2 about reframing the sequence of iterates to achieve a more interpretable algorithm. We defer the proof to the full version.

▶ **Lemma 29** (Connecting Algorithm 4 to Algorithm 2). *Consider Algorithm 2 with input given by a $(\psi, B_R^n(x_0), \mathbb{R}^n, f^*)$-unbounded setup (Definition 10); $y_0 := \nabla f(x_0)$; and K, $(\epsilon_k)_{k=1}^K$, and $(\lambda_k)_{k=0}^{K-1}$ as in Algorithm 4. Then letting $(z_k)_{k=0}^K$ denote the sequence of iterates generated by Algorithm 4, the following are valid sequences of iterates for Algorithm 2:*

$$q_k = \nabla f\left(\frac{1}{\Lambda_k}\sum_{j=0}^{k-1} \lambda_j z_j\right) \quad \text{for } k \in [K], \tag{12}$$

$$x_k = (1+\Lambda_k) z_k - \sum_{j=0}^{k-1} \lambda_j z_j \quad \text{for } k \in [K], \text{ and} \tag{13}$$

$$y_k = \nabla f(z_k) \quad \text{for } k \in \{0\} \cup [K]. \tag{14}$$

Having connected Algorithm 4 to Algorithm 2, we can apply the schedules given in Corollary 17 to show that Algorithm 4 returns a γ-critical point of h with an optimal complexity. We defer the proof to the full version.

▶ **Theorem 30** (Guarantee for Algorithm 4). *For any[6] $\gamma \in (0, \sqrt{2\beta\Delta})$ and with $K = O(\log(\beta\Delta/\gamma))$, the output of Algorithm 4 with schedules given by*

$$\lambda_k = \frac{2^k}{32} \text{ for } k \in \{0\} \cup [K-1] \text{ and } \epsilon_k = \frac{\Delta}{64 \cdot 1.5^k} \text{ for } k \in [K] \tag{15}$$

satisfies $\|\nabla h(z_K)\| \le \gamma$, and the algorithm makes at most $O\left(\frac{\sqrt{\beta\Delta}}{\gamma}\right)$ gradient queries to h.

[6] The restriction on γ is without loss of generality since $\|\nabla h(x_0)\| \le \sqrt{2\beta\Delta}$ by smoothness. We add it because it simplifies the analysis.

References

1 Jacob D Abernethy and Jun-Kun Wang. On frank-wolfe and equilibrium computation. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

2 Ilan Adler. The equivalence of linear programs and zero-sum games. *International Journal of Games Theory*, Volume 42:165-177, February 2013.

3 Zeyuan Allen-Zhu and Elad Hazan. Optimal black-box reductions between optimization objectives. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 1614–1622, Red Hook, NY, USA, 2016. Curran Associates Inc.

4 Hilal Asi, Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Stochastic bias-reduced gradient methods. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2024. Curran Associates Inc.

5 Yair Carmon and Danielle Hausler. Distributionally robust optimization via ball oracle acceleration. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc.

6 Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Thinking inside the ball: Near-optimal minimization of the maximal loss. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 866–882. PMLR, 15–19 August 2021. URL: `http://proceedings.mlr.press/v134/carmon21a.html`.

7 Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Recapp: Crafting a more efficient catalyst for convex optimization. In *International Conference on Machine Learning*, 2022.

8 Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. *A Whole New Ball Game: A Primal Accelerated Method for Matrix Games and Minimizing the Maximum of Smooth Functions*, pages 3685–3723. Society for Industrial and Applied Mathematics, 2024. `doi:10.1137/1.9781611977912.130`.

9 Yair Carmon, Yujia Jin, Aaron Sidford, and Kevin Tian. Variance reduction for matrix games. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

10 Kenneth L. Clarkson, Elad Hazan, and David P. Woodruff. Sublinear optimization for machine learning. *J. ACM*, 59(5), November 2012. `doi:10.1145/2371656.2371658`.

11 Michael B. Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 938–942, New York, NY, USA, 2019. Association for Computing Machinery. `doi:10.1145/3313276.3316303`.

12 Michael B. Cohen, Aaron Sidford, and Kevin Tian. Relative lipschitzness in extragradient methods and a direct recipe for acceleration. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 62:1–62:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPICS.ITCS.2021.62`.

13 Sebastian Curi, Kfir Y. Levy, Stefanie Jegelka, and Andreas Krause. Adaptive sampling for stochastic risk-averse learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

14 G. B. Dantzig. Linear programming and extensions, 1953.

15 Jelena Diakonikolas and Puqian Wang. Potential function-based framework for minimizing gradients in convex and min-max optimization. *SIAM Journal on Optimization*, 32(3):1668–1697, 2022. `doi:10.1137/21M1395302`.

16 John Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49, June 2021.

17 I. Ekeland, Roger Temam, Society for Industrial, and Applied Mathematics. *Convex analysis and variational problems*. Classics in applied mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa., 1999.

18 Roy Frostig, Rong Ge, Sham M. Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 2540–2548. JMLR.org, 2015. URL: `http://proceedings.mlr.press/v37/frostig15.html`.

19 Michael B. Giles. Multilevel monte carlo path simulation. *Operations Research*, 56(3):607–617, June 2008. `doi:10.1287/OPRE.1070.0496`.

20 Michael B. Giles. Multilevel monte carlo methods. *Acta Numerica*, 24:259–328, May 2015. `doi:10.1017/S096249291500001X`.

21 G. N. Grapiglia and Yurii Nesterov. Tensor methods for finding approximate stationary points of convex functions. *Optimization Methods and Software*, 37(2):605–638, 2022. `doi:10.1080/10556788.2020.1818082`.

22 Michael D. Grigoriadis and Leonid G. Khachiyan. A sublinear-time randomized approximation algorithm for matrix games. *Operations Research Letters*, 18(2):53–58, 1995. `doi:10.1016/0167-6377(95)00032-0`.

23 Yujia Jin, Aaron Sidford, and Kevin Tian. Sharper rates for separable minimax and finite sum optimization via primal-dual extragradient methods. In Po-Ling Loh and Maxim Raginsky, editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 4362–4415. PMLR, 02–05 July 2022. URL: `https://proceedings.mlr.press/v178/jin22b.html`.

24 Donghwan Kim and Jeffrey A. Fessler. Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions. *Journal of Optimization Theory and Applications*, 188(1):192–219, January 2021. `doi:10.1007/S10957-020-01770-2`.

25 Jaeyeon Kim, Asuman Ozdaglar, Chanwoo Park, and Ernest K. Ryu. Time-reversed dissipation induces duality between minimizing gradient norm and function value, 2023. `arXiv:2305.06628`.

26 Jaeyeon Kim, Chanwoo Park, Asuman Ozdaglar, Jelena Diakonikolas, and Ernest K. Ryu. Mirror duality in convex optimization, 2024. `arXiv:2311.17296`.

27 Guanghui Lan, Yuyuan Ouyang, and Zhe Zhang. Optimal and parameter-free gradient minimization methods for convex and nonconvex optimization, 2023. `arXiv:2310.12139`.

28 Jongmin Lee, Chanwoo Park, and Ernest K. Ryu. A geometric structure of acceleration and its role in making gradients small fast. In *Neural Information Processing Systems*, 2021.

29 Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8847–8860. Curran Associates, Inc., 2020.

30 Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 3384–3392, Cambridge, MA, USA, 2015. MIT Press.

31 M. Minsky and S. Papert. Perceptrons: An introduction to computational geometry, 1988.

32 Hongseok Namkoong and John C. Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 2216–2224, Red Hook, NY, USA, 2016. Curran Associates Inc.

33 A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009. `doi:10.1137/070704277`.

34 A.S Nemirovsky. On optimality of krylov's information when solving linear operator equations. *Journal of Complexity*, 7(2):121–130, 1991. `doi:10.1016/0885-064X(91)90001-E`.

35 A.S Nemirovsky. Information-based complexity of linear operator equations. *Journal of Complexity*, 8(2):153–175, 1992. doi:10.1016/0885-064X(92)90013-2.

36 Yurii Nesterov. *Lectures on Convex Optimization*. Springer Publishing Company, Incorporated, 2nd edition, 2018.

37 Yurii Nesterov, Alexander Gasnikov, Sergey Guminov, and Pavel Dvurechensky. Primal-dual accelerated gradient methods with small-dimensional relaxation oracle. *Optimization Methods and Software*, 36(4):773–810, July 2021. doi:10.1080/10556788.2020.1731747.

38 Francesco Orabona. A modern introduction to online learning, 2023. arXiv 1912.13213. arXiv:1912.13213.

39 Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8:171–176, 1958.

40 Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization, 2008.

41 J. v. Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, December 1928.

42 Jan van den Brand, Yin Tat Lee, Yang P. Liu, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Minimum cost flows, mdps, and ℓ_1-regression in nearly linear time for dense instances. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 859–869, New York, NY, USA, 2021. Association for Computing Machinery.

43 Jun-Kun Wang, Jacob Abernethy, and Kfir Y. Levy. No-regret dynamics in the fenchel game: a unified framework for algorithmic convex optimization. *Mathematical Programming*, 205(1-2):203–268, May 2024. doi:10.1007/S10107-023-01976-Y.

Provability of the Circuit Size Hierarchy and Its Consequences

Marco Carmosino
IBM Research, Cambridge, MA, USA

Valentine Kabanets
Simon Fraser University, Burnaby, Canada

Antonina Kolokolova
Memorial University of Newfoundland, St. John, Canada

Igor C. Oliveira
University of Warwick, UK

Dimitrios Tsintsilidas
University of Warwick, UK

Abstract

The *Circuit Size Hierarchy* (CSH^a_b) states that if $a > b \geq 1$ then the set of functions on n variables computed by Boolean circuits of size n^a is strictly larger than the set of functions computed by circuits of size n^b. This result, which is a cornerstone of circuit complexity theory, follows from the *non-constructive* proof of the existence of functions of large circuit complexity obtained by Shannon in 1949.

Are there more "constructive" proofs of the Circuit Size Hierarchy? Can we quantify this? Motivated by these questions, we investigate the provability of CSH^a_b in theories of bounded arithmetic. Among other contributions, we establish the following results:

- **(i)** Given any $a > b > 1$, CSH^a_b is provable in Buss's theory T^2_2.
- **(ii)** In contrast, if there are constants $a > b > 1$ such that CSH^a_b is provable in the theory T^1_2, then there is a constant $\varepsilon > 0$ such that P^{NP} requires non-uniform circuits of size at least $n^{1+\varepsilon}$.

In other words, an improved *upper bound* on the proof complexity of CSH^a_b would lead to new *lower bounds* in complexity theory.

We complement these results with a proof of the *Formula Size Hierarchy* (FSH^a_b) in PV_1 with parameters $a > 2$ and $b = 3/2$. This is in contrast with typical formalizations of complexity lower bounds in bounded arithmetic, which require APC_1 or stronger theories and are not known to hold even in T^1_2.

2012 ACM Subject Classification Theory of computation → Complexity theory and logic; Theory of computation → Proof complexity; Theory of computation → Circuit complexity; Theory of computation → Proof theory

Keywords and phrases Bounded Arithmetic, Circuit Complexity, Hierarchy Theorems

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.30

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2024/145/`

Funding This work received support from the Royal Society University Research Fellowship URF\R1\191059; the UKRI Frontier Research Guarantee Grant EP/Y007999/1; the Centre for Discrete Mathematics and its Applications (DIMAP) at the University of Warwick, and the Natural Sciences and Engineering Research Council of Canada.

Acknowledgements We thank Emil Jeřábek for a discussion about witnessing theorems in bounded arithmetic. We are also grateful to Hanlin Ren for a suggestion that improved our bounds in Corollary 10.



© Marco Carmosino, Valentine Kabanets, Antonina Kolokolova, Igor C. Oliveira, and Dimitrios Tsintsilidas;
licensed under Creative Commons License CC-BY 4.0



1 Introduction

1.1 Context and Motivation

The existence of Boolean functions requiring large circuits can be shown by a non-constructive counting argument, as established by Shannon in 1949 [24]. It follows from Shannon's seminal result and a simple padding argument that if $a > b \geq 1$ there are functions computable by circuits of size n^a that cannot be computed by circuits of size n^b. In other words, the classification of Boolean functions by their minimum circuit size forms a strict *hierarchy*.

Obtaining a "constructive" form of these results has been a holy grail in computational complexity theory for several decades due to its connections to derandomization and as an approach to separating P and NP. For instance, if there is a polynomial-time algorithm that given 1^n outputs the truth-table of a function $f\colon \{0,1\}^{\log n} \to \{0,1\}$ that requires circuits of size $n^{\Omega(1)}$, then $\mathsf{P} = \mathsf{BPP}$ [9]. In results of this form, a constructive form of the (non-constructive) proof of the existence of hard functions is interpreted *computationally* as the existence of an algorithm of bounded complexity that computes a hard function.

In this paper, rather than focusing on the existence of algorithms to capture the constructiveness of a statement, we explore this notion from the perspective of mathematical logic, specifically concerning its *provability* in certain mathematical theories. We are interested in identifying the weakest theory capable of establishing the aforementioned circuit size hierarchy for Boolean circuits and related results.

As one of our contributions, we present a tight connection between the computational and proof-theoretic perspectives. We demonstrate that proving the non-uniform circuit size hierarchy in a theory known as T^1_2 implies the existence of a function in P^{NP} that requires Boolean circuits of size at least $n^{1+\varepsilon}$. The latter is a frontier question in complexity theory (see, e.g., [5]). Thus, in a precise sense, developing more constructive proofs of the circuit size hierarchy would lead to significant progress on explicit circuit lower bounds.

We now proceed to describe this result and other contributions of this work in detail.

1.2 Results

We will be concerned with standard theories of bounded arithmetic. These theories are designed to capture proofs that manipulate and reason with concepts from a specified complexity class. Notable examples include Cook's theory PV_1 [7], which formalizes polynomial-time reasoning; Jeřábek's theory APC_1 [10, 11, 13], which extends PV_1 by incorporating the dual weak pigeonhole principle for polynomial-time functions and formalizes probabilistic polynomial-time reasoning; and Buss's theories T^i_2 [2], which incorporate induction principles corresponding to various levels of the polynomial-time hierarchy.

For an introduction to bounded arithmetic, we refer to [3]. For its connections to computational complexity and a discussion on the formalization of complexity theory, we refer to [23].[1] Here we only recall that theory PV_1 corresponds essentially to T^0_2 [12], and that $\mathsf{T}^0_2 \subseteq \mathsf{T}^1_2 \subseteq \mathsf{T}^2_2$ correspond to the first levels of Buss's hierarchy. A brief overview of the theories is provided in Section 2.

For a given $n \in \mathbb{N}$, we use $\mathsf{CIRCUIT}[s(n)]$ to denote the set of Boolean functions $f\colon \{0,1\}^n \to \{0,1\}$ computed by circuits of size at most $s(n)$. Similarly, when referring to formula size, we write $\mathsf{FORMULA}[s(n)]$. We use $\mathsf{SIZE}[s(n)]$ to denote the set of languages $L \subseteq \{0,1\}^*$ that admit a sequence of circuits of size at most $s(n)$.

[1] In particular, the reference [23] contains a detailed discussion of some aspects of the formalization of the statements appearing below.

Circuit Size Hierarchy

For rationals $a > b \geq 1$ and n_0, we consider the following sentence:[2]

$$\mathsf{CSH}[a,b,n_0] \equiv \forall n \geq n_0 \in \mathsf{Log},\ \exists \text{ circuit } D\colon \{0,1\}^n \to \{0,1\} \text{ of size} \leq n^a,$$
$$\forall \text{ circuit } C\colon \{0,1\}^n \to \{0,1\} \text{ of size} \leq n^b,\ \exists x \in \{0,1\}^n \text{ such that } D(x) \neq C(x).$$

In other words, $\mathsf{CSH}[a,b,n_0]$ states that $\mathsf{CIRCUIT}[n^a] \not\subseteq \mathsf{CIRCUIT}[n^b]$ whenever $n \geq n_0$.

Next, we state our first result.

▶ **Theorem 1.** *The following results hold:*

(i) *For every choice of rationals a and b with $a > b > 1$, and for every large enough $n_0 \in \mathbb{N}$,*

$$\mathsf{T}^2_2 \vdash \mathsf{CSH}[a,b,n_0]\,.$$

(ii) *If there are rationals $a > b > 1$ and a constant $n_0 \in \mathbb{N}$ such that*

$$\mathsf{T}^1_2 \vdash \mathsf{CSH}[a,b,n_0]\,,$$

then there is a constant $\varepsilon > 0$ and a language $L \in \mathsf{P}^{\mathsf{NP}}$ such that $L \notin \mathsf{SIZE}[n^{1+\varepsilon}]$.

(iii) *Similarly to the previous item, if $\mathsf{PV}_1 \vdash \mathsf{CSH}[a,b,n_0]$, there is $L \in \mathsf{P}$ such that $L \notin \mathsf{SIZE}[n^{1+\varepsilon}]$.*

To put it another way, we can establish a circuit size hierarchy within the theory T^2_2. If this result could also be proven in the theory T^1_2, it would lead to a significant breakthrough in circuit lower bounds. Thus, by enhancing the proof complexity upper bound for the provability of the circuit size hierarchy, we can achieve new circuit lower bounds.

The proof technique of Item (ii) also applies to the theory T^2_2, which combined with Item (i) gives us a superlinear lower bound for a language in $\mathsf{P}^{\Sigma^p_2}$, but this is already known by Kannan's theorem [15].

Note that in Theorem 1 Items (ii) and (iii) we obtain a lower bound against circuits of size $n^{1+\varepsilon}$, where the constant $\varepsilon > 0$ depends on the proof of $\mathsf{CSH}[a,b,n_0]$ in the corresponding theory. In other words, while the sentence claims the existence of hardness against circuits of size n^b, we are only able to extract a weaker lower bound for an explicit problem.

In our next result, we describe a setting where we can extract all the hardness from a proof of the corresponding sentence.

Succinct Circuit Size Hierarchy

We define what we call the succinct version of the circuit size hierarchy, where we substitute the upper bound circuit with a collection of labelled examples for the function, which can always represent a circuit. For rationals $a > b \geq 1$ and n_0, we consider the following sentence:

$$\begin{aligned}\mathsf{SCSH}[a,b,n_0] \equiv\ & \forall n \geq n_0 \in \mathsf{Log},\ \exists \text{ collection } \{(x^1,b^1),\ldots,(x^\ell,b^\ell)\} \text{ of size } \ell \leq n^a \text{ with}\\ & |x^i| = n \wedge |b^i| = 1 \text{ for each } i \in [\ell] \text{ and } x^i \neq x^j \text{ for distinct } i,j \in [\ell]\,,\\ & \forall \text{ circuit } C\colon \{0,1\}^n \to \{0,1\} \text{ of size} \leq n^b,\ \exists i \in [\ell] \text{ s.t. } C(x^i) \neq b^i.\end{aligned}$$

In other words, $\mathsf{SCSH}[a,b,n_0]$ states that for every $n \geq n_0$ there is a collection of $\ell \leq n^a$ labelled examples such that every circuit of size at most n^b disagrees with at least one of its labels. The truth of this statement can be validated by a counting argument, similarly with the circuit size hierarchy proof.

[2] The abbreviation $n \in \mathsf{Log}$ denotes that n is the length of a variable N (see, e.g., [23] for more details).

We obtain the following results on the proof complexity of the succinct circuit size hierarchy.

▶ **Theorem 2.** *The following results hold:*

(i) *For every choice of rationals $a > b > 1$ and for every large enough $n_0 \in \mathbb{N}$,*

$$\mathsf{T}^2_2 \vdash \mathsf{SCSH}[a, b, n_0]\,.$$

(ii) *If there are rationals $a > b > 1$ and a constant $n_0 \in \mathbb{N}$ such that*

$$\mathsf{T}^1_2 \vdash \mathsf{SCSH}[a, b, n_0]\,,$$

then there is a language $L \in \mathsf{P}^{\mathsf{NP}}$ such that $L \notin \mathsf{SIZE}[n^b]$.

In our final result, we investigate the provability of size hierarchies for more restricted computational models in T^1_2 and weaker theories.

Formula Size Hierarchy

For rationals $a > b \geq 1$ and n_0, we consider the following sentence:

$$\mathsf{FSH}[a, b, n_0] \equiv \forall n \geq n_0 \in \mathsf{Log},\ \exists \text{ formula } F\colon \{0,1\}^n \to \{0,1\} \text{ of size} \leq n^a,$$
$$\forall \text{ formula } G\colon \{0,1\}^n \to \{0,1\} \text{ of size} \leq n^b,\ \exists x \in \{0,1\}^n \text{ such that } F(x) \neq G(x).$$

In other words, $\mathsf{FSH}(a, b, n_0)$ states that $\mathsf{FORMULA}[n^a] \not\subseteq \mathsf{FORMULA}[n^b]$ whenever $n \geq n_0$.

We establish that for some parameters a formula size hierarchy is provable already in PV_1.

▶ **Theorem 3.** *Consider rationals $a > 2$ and $b = 3/2$, and let n_0 be a large enough positive integer. Then*

$$\mathsf{PV}_1 \vdash \mathsf{FSH}[a, b, n_0]\,.$$

While many lower bounds can be proven in APC_1 and stronger theories (see [22, 23, 4] and references therein), Theorem 3 provides an example of a non-trivial lower bound (under a "Log" formalization; see [23, Section 4.1]) that can be established in PV_1, which might be of independent interest.

1.3 Techniques

The proofs of Items (ii) and (iii) in Theorem 1 are inspired by arguments from [18, 17] that rely on a combination of a witnessing theorem with a term elimination strategy. Recall that the witnessing theorem allows us to extract computational information from a proof of the sentence in the theory. Roughly speaking, in our context this implies that the first existential quantifier in the sentence $\mathsf{CSH}[a, b, n_0]$, which corresponds to a circuit computing a hard function, can be witnessed by a finite number of terms $t_1, \ldots, t_k$ of the corresponding theory. In PV_1, a term yields a polynomial-time function, while in T^1_2 a term yields a polynomial-time function with access to an NP oracle. The main difficulty is that (1) for a given input length n it is not clear which term among $t_1, \ldots, t_k$ succeeds in constructing a hard function, and (2) for a term to succeed we must provide counter-examples to the candidate witnesses provided by previous terms.

As in previous papers, we assume that the conclusion of the theorem does not hold, and use this assumption to rule out the correctness of each term. This leads to a contradiction, meaning that the original sentence is not provable in the corresponding theory. Implementing

this plan requires a careful argument, and we are currently only able to carry it out under a complexity inclusion in $\mathsf{SIZE}[n^{1+\varepsilon}]$ as opposed to $\mathsf{SIZE}[n^b]$. The proof of the result is given in Section 3.1.

On the other hand, in the case of the succinct circuit size hierarchy, the argument for Item (ii) of Theorem 2 is simpler and allows us to start with the weaker assumption that $\mathsf{P}^{\mathsf{NP}} \subseteq \mathsf{SIZE}[n^b]$. Without getting into the technical details, the main reason for not losing hardness in this result is that given a labelled list of examples and access to an NP oracle, we can efficiently compute a minimum size circuit that agrees with this list of inputs. Consequently, we can check if a candidate labelled list provided by a term is indeed hard, or produce a counter-example when this is not the case. The same computation is not available in the case of Theorem 1, since it is not clear how to efficiently compute with access to an NP oracle if a given circuit admits a smaller equivalent circuit. The proof of Item (ii) of Theorem 2 appears in Section 3.2.

The proofs of Theorem 1 Item (i) and Theorem 2 Item (i) are given in Section 3.3. The formalization of these hierarchies in T^2_2 is easily done with access to the dual Weak Pigeonhole Principle for polynomial-time functions, a principle which is known to be available in T^2_2. In more detail, CSH follows from SCSH in PV_1, while SCSH can be established in theory APC_1, which is contained in T^2_2.

Finally, in the proof of Theorem 3 we formalize in PV_1 that the parity function on n bits can be computed by formulas of size $O(n^2)$ and require formulas of size $\Omega(n^{3/2})$. This yields in PV_1 a proof of $\mathsf{FSH}[a, b, n_0]$ for any choice of parameters $a > 2$, large enough n_0, and $b = 3/2$. The upper bound on the complexity of parity follows from a straightforward formalization of the correctness of the formula obtained via a divide-and-conquer procedure. On the other hand, in order to show the formula lower bound we formalize Subbotovskaya's argument [25] based on the method of restrictions. To implement the proof in PV_1, we directly define an efficient refuter that given a small formula outputs an input string where it fails to compute the parity function. The correctness of the refuter is established by induction using an induction principle available in the theory S^1_2. We then rely on a conservation result showing that the proof can also be done in PV_1. A detailed exposition of the argument appears in Section 4.

2 Preliminaries

2.1 Complexity Theory

We employ standard definitions from complexity theory, such as basic complexity classes, Boolean circuits, and Boolean formulas (see, e.g., [1]).

Let $\mathbb{N}$ represent the set of non-negative integers. For any $a \in \mathbb{N}$, let $|a|$ denote the length of its binary representation, defined as $|a| \triangleq \lceil \log_2(a+1) \rceil$. For a constant $k \geq 1$, a function $f\colon \mathbb{N}^k \to \mathbb{N}$ is said to be computable in polynomial time if $f(x_1, \ldots, x_k)$ can be computed in time polynomial in $|x_1|, \ldots, |x_k|$. For convenience, we might write $|\vec{x}| \triangleq |x_1|, \ldots, |x_k|$. The class FP denotes the set of polynomial-time computable functions. Although the definition of polynomial time typically refers to a machine model, FP can also be defined in a machine-independent manner as the closure of a set of base functions $\mathcal{F}$ (not described here) under *composition* and *limited recursion on notation*. A function $f(\vec{x}, y)$ is defined from functions $g(\vec{x})$, $h(\vec{x}, y, z)$, and $k(\vec{x}, y)$ by *limited recursion on notation* if

$$f(\vec{x}, 0) = g(\vec{x})$$
$$f(\vec{x}, y) = h(\vec{x}, y, f(\vec{x}, \lfloor y/2 \rfloor))$$
$$f(\vec{x}, y) \leq k(\vec{x}, y)$$

for every sequence $(\vec{x}, y)$ of natural numbers. Cobham [6] established that FP is the smallest class of functions that contains the base functions $\mathcal{F}$ and is closed under composition and limited recursion on notation.

2.2 Bounded Arithmetic

2.2.1 Logical Theories

We recall the definitions of some standard theories of bounded arithmetic. For more details, the reader can consult [16, 8, 20].

2.2.1.1 Cook's Theory PV [7]

The theory PV_1 is designed to model the set $\mathbb{N}$ of natural numbers with the standard interpretations for constants and function symbols like $0, +, \times$, etc. The vocabulary (language) of PV, denoted $\mathcal{L}_{\mathsf{PV}}$, includes a function symbol for each polynomial-time algorithm $f\colon \mathbb{N}^k \to \mathbb{N}$, where k is any constant. These function symbols and their defining axioms are derived using Cobham's characterization of polynomial-time functions discussed above. While Cook's PV was an equational theory, it was later extended in [19] to a first-order theory PV_1, which includes an induction axiom scheme that simulates binary search. It can be shown that PV_1 allows induction over quantifier-free formulas (i.e., polynomial-time predicates).

PV_1 can be formulated with all axioms as universal formulas (i.e., $\forall \vec{x}\, \phi(\vec{x})$, where ϕ is free of quantifiers). Thus, PV_1 is a *universal theory*. Although the definition of PV_1 is quite technical, the theory is fairly robust and the details of its definition are often unnecessary for practical purposes. In particular, PV_1 has an equivalent formalizations that does not rely on Cobham's result, e.g. [12].

2.2.1.2 Jeřábek's Theory APC_1 [10, 11, 13]

APC_1 extends PV_1 with the *dual Weak Pigeonhole Principle* (dWPHP) for PV_1 functions:

$$\mathsf{APC}_1 \triangleq \mathsf{PV} \cup \{\mathsf{dWPHP}(f) \mid f \in \mathcal{L}_{\mathsf{PV}}\}.$$

Each sentence $\mathsf{dWPHP}(f)$ postulates that, for every length $n = |N|$ and for every choice of $\vec{z}$, there is $y < (1 + 1/n) \cdot 2^n$ such that $f(\vec{z}, x) \neq y$ for every $x < 2^n$. It is known that APC_1 is contained in T^2_2 [21].

2.2.1.3 Buss's Theories S^i_2 and T^i_2 [2]

The language $\mathcal{L}_B$ for these theories includes predicate symbols $=$ and $\leq$, constant symbols 0 and 1, and function symbols S (successor), $+$, $\cdot$, $\lfloor x/2 \rfloor$, $|x|$ (interpreted as the length of x), and $\#$ (interpreted as $x\#y = 2^{|x|\cdot|y|}$, known as "smash").

Recall that a *bounded quantifier* is a quantifier of the form $Qy \leq t$, where $Q \in \{\exists, \forall\}$ and t is a term not involving y. Similarly, a *sharply bounded quantifier* is one of the form $Qy \leq |t|$. A formula where each quantifier appears bounded (or sharply bounded) is called a bounded (or sharply bounded) formula.

We can create a hierarchy of formulas by counting alternations of bounded quantifiers. The class $\Pi^b_0 = \Sigma^b_0$ contains the sharply bounded formulas. Recursively, for each $i \geq 0$, the classes Σ^b_i and Π^b_i are defined by the quantifier structure of the sentence, ignoring sharply bounded quantifiers. For instance, if $\varphi \in \Sigma^b_0$ and $\psi \triangleq \exists y \leq t(\vec{x})\ \varphi(y, \vec{x})$, then $\psi \in \Sigma^b_1$. For

the general case of the definition, see [16]. It is known that for each $i \geq 1$, a predicate $P(\vec{x})$ is in Σ_i^p (the i-th level of the polynomial hierarchy) if and only if there is a Σ_i^b-formula that agrees with it over $\mathbb{N}$.

These theories share a common set of finitely many axioms, BASIC, which postulate the expected arithmetic behavior of the constants, predicates, and function symbols. The only difference among the theories is the type of induction axiom scheme each one postulates.

T_2^i is a theory in the language $\mathcal{L}_B$ that extends BASIC by including the induction axiom IND:

$$\varphi(0) \wedge \forall x\,(\varphi(x) \to \varphi(x+1)) \to\ \forall x\,\varphi(x)$$

for all Σ_i^b-formulas $\varphi(a)$. The formula $\varphi(a)$ may contain other free variables in addition to a.

S_2^i is a theory in the language $\mathcal{L}_B$ that extends BASIC by including the polynomial induction axiom PIND:

$$\varphi(0) \wedge \forall x\,(\varphi(\lfloor x/2 \rfloor) \to \varphi(x)) \to\ \forall x\,\varphi(x)$$

for all Σ_i^b-formulas $\varphi(a)$. The formula $\varphi(a)$ may contain other free variables in addition to a.

2.2.1.4 Theory $\mathsf{S}_2^1(\mathsf{PV})$

When proving some results in S_2^1, it is often convenient to use a more expressive vocabulary that easily describes any polynomial-time function. This can be done in a *conservative* manner, meaning the power of the theory is not increased. Specifically, let Γ be a set of $\mathcal{L}_B$-formulas. We say that a polynomial-time function $f\colon \mathbb{N}^k \to \mathbb{N}$ is Γ-*definable* in S_2^1 if there exists a formula $\psi(\vec{x}, y) \in \Gamma$ such that the following conditions are met:

(i) For every $\vec{a} \in \mathbb{N}^k$, $f(\vec{a}) = b$ if and only if $\mathbb{N} \models \varphi(\vec{a}, b)$.

(ii) $\mathsf{S}_2^1 \vdash \forall \vec{x}\,(\exists y\,(\varphi(\vec{x}, y) \wedge \forall z\,(\varphi(\vec{x}, z) \to y = z)))$.

Every function $f \in \mathsf{FP}$ is Σ_1^b-definable in S_2^1. By incorporating all functions in FP into the vocabulary of S_2^1 and extending the axioms of S_2^1 with their defining equations, we obtain a theory $\mathsf{S}_2^1(\mathsf{PV})$. This theory allows polynomial-time predicates to be referred to using quantifier-free formulas. $\mathsf{S}_2^1(\mathsf{PV})$ remains conservative over S_2^1, meaning any $\mathcal{L}_B$-sentence provable in $\mathsf{S}_2^1(\mathsf{PV})$ is also provable in S_2^1. Finally, it is known that $\mathsf{S}_2^1(\mathsf{PV})$ proves the polynomial induction scheme for both Σ_1^b-formulas and Π_1^b-formulas within the extended vocabulary.

2.2.2 The KPT Witnessing Theorem

The following witnessing theorem (a variant of Herbrand's theorem) is proved in [19] (cf. also [16, Theorem 7.4.1]) for universal theories (like the theory PV_1).

▶ **Theorem 4** (KPT Theorem for $\forall\exists\forall\exists$ sentences)**.** *Let* T *be a universal theory with vocabulary* $\mathcal{L}$*. Let* φ *be an open* $\mathcal{L}$*-formula, and suppose that*

$$\mathsf{T} \vdash\ \forall x\,\exists y\,\forall z\,\exists w\ \varphi(x, y, z, w).$$

Then there is a finite sequence $s_1, \ldots, s_k$ *of* $\mathcal{L}$*-terms such that*

$$\mathsf{T} \vdash\ \forall x, z_1, \ldots, z_k\ \big(\psi(x, s_1(x), z_1) \vee \psi(x, s_2(x, z_1), z_2) \vee \cdots \vee \psi(x, s_k(x, z_1, \ldots, z_{k-1}), z_k)\big),$$

where

$$\psi(x, y, z) \triangleq \exists w\ \varphi(x, y, z, w).$$

We can also apply the KPT Theorem to each theory T^i_2 (for $i \geq 1$) using a conservative extension of the theory that admits a universal axiomatization. The corresponding theory is called PV_{i+1} [19]. In PV_{i+1}, each term is equivalent to an $\mathsf{FP}^{\Sigma^p_i}$ function over the standard model. This leads to the following result.

▶ **Theorem 5** (Consequence of the KPT Theorem for Theory T^i_2)**.** *Let $i \geq 1$, $\varphi(x, y, w, z)$ be a Π^b_i-formula, and suppose that*

$$\mathsf{T}^i_2 \vdash \forall x\, \exists y \forall z\, \exists w\, \varphi(x, y, w, z).$$

Then there is a finite sequence $f_1, \ldots, f_k$ of function symbols, each corresponding to an $\mathsf{FP}^{\Sigma^p_i}$ function, such that

$$\mathbb{N} \models \forall x, z_1, \ldots, z_k \left(\psi(x, f_1(x), z_1) \vee \psi(x, f_2(x, z_1), z_2) \vee \cdots \vee \psi(x, f_k(x, z_1, \ldots, z_{k-1}), z_k)\right),$$

where

$$\psi(x, y, z) \triangleq \exists w\, \varphi(x, y, z, w).$$

3 Circuit Size Hierarchies in Bounded Arithmetic

3.1 Explicit Circuit Lower Bounds from Provability in PV_1 and T^1_2

In this section, we prove Theorem 1 Items (ii) and Items (iii).

▶ **Theorem 6** (Theorem 1 Item (iii))**.** *If there are rationals $a > b > 1$ and $n_0 \in \mathbb{N}$ such that*

$$\mathsf{PV}_1 \vdash \mathsf{CSH}[a, b, n_0]\,,$$

then there is a constant $\varepsilon > 0$ and a language $L \in \mathsf{P}$ such that $L \notin \mathsf{SIZE}[n^{1+\varepsilon}]$.

Proof. Towards a contradiction, suppose that $\mathsf{PV}_1 \vdash \mathsf{CSH}[a, b, n_0]$ for rationals $a > b > 1$ and some constant n_0 and that $P \subseteq \bigcap_{\varepsilon>0} \mathsf{SIZE}[n^{1+\varepsilon}]$. The sentence $\mathsf{CSH}[a, b, n_0]$ has the form $\forall\exists\forall\exists$:

$$\mathsf{CSH}[a, b, n_0] \triangleq \forall n \geq n_0 \in \mathsf{Log},\ \exists \text{ circuit } D\ \forall \text{ circuit } C\ \ \psi_{a,b}(n, D, C)\,,$$

where $\psi_{a,b}(n, D, C)$ is the existential formula:

$$\psi_{a,b}(n, D, C) \triangleq \exists x\ |x| \leq n \wedge \mathsf{SIZE}(D) \leq n^a \wedge (\mathsf{SIZE}(C) \leq n^b \rightarrow\ D(x) \neq C(x)).$$

Therefore, we can apply the KPT Theorem (Theorem 4), which provides PV_1-terms, equivalently FP functions, $s_1, \ldots, s_k$, where k is a constant, such that

$$\mathbb{N} \models \psi_{a,b}(n, s_1(1^{(n)}), C_1) \vee \psi_{a,b}(n, s_2(1^{(n)}, C_1), C_2) \vee \cdots \vee \psi_{a,b}(n, s_k(1^{(n)}, C_1, \ldots, C_{k-1}), C_k). \tag{1}$$

In the formula above the circuits $C_1, \ldots, C_k$ are universally quantified.

Next, we use $P \subseteq \bigcap_{\varepsilon>0} \mathsf{SIZE}[n^{1+\varepsilon}]$ to refute each of these disjuncts. We start by considering the following language, D-Eval:

D-Eval is in P due to the fact that $s_1, \ldots, s_k \in \mathsf{FP}$ and circuit evaluation is in FP. By our assumption on the circuit complexity of the complexity class P, for every input length m and every $\varepsilon > 0$, D-Eval $\in \mathsf{SIZE}[m^{1+\varepsilon}]$, so we can choose

$$\varepsilon_0 \triangleq b^{1/(2k)} - 1 > 0$$

Algorithm 1 The pseudocode of an algorithm that decides the language D-Eval.

Input : A string x and a sequence $\langle C_1, C_2, \ldots, C_r \rangle$ of $r \leq k-1$ circuits
Define $n \triangleq |x|$;
Simulate $s_{r+1}(1^{(n)}, C_1, \ldots, C_r)$ and interpret the output as a Boolean circuit $D\colon \{0,1\}^n \to \{0,1\}$;
// We assume w.l.o.g. that D is a valid n-bit circuit of size $\leq n^a$, since otherwise the disjunct is trivially false.
Evaluate D on input x and output the result.

and have D-Eval $\in \mathsf{SIZE}[m^{b^{1/(2k)}}]$. We also define the constants

$$\epsilon_i \triangleq b^{i/k} \quad \text{and} \quad \delta_i \triangleq b^{(2i-1)/(2k)}$$

for $i = 1, \ldots, k$. Note that $\epsilon_i = (1+\varepsilon_0)\delta_i$ and $\delta_{i+1} > \epsilon_i$.

We start by refuting $\psi_{a,b}(n, s_1(1^{(n)}), C_1)$. We consider inputs of the form x, λ to D-Eval, where λ is the empty sequence. Then the input has length $n + c$, where $c = O(\log n)$ accounts for the overhead in the encoding of the input. We consider the circuit $C_1^* \in \mathsf{CIRCUIT}[(n+c)^{1+\varepsilon_0}]$, which evaluates as D-Eval on inputs of length $n+c$, and we fix the input variables not related to x to represent the empty sequence. The resulting circuit has as input an n-bit string x and computes according to $s_1(1^{(n)})$ by definition of the D-Eval algorithm. For sufficiently large n, we have that $n + c \leq n^{\delta_1} \Rightarrow (n+c)^{1+\varepsilon_0} \leq n^{(1+\varepsilon_0)\delta_1} = n^{\epsilon_1}$, therefore we have the circuit $C_1^* \in \mathsf{CIRCUIT}[n^{\epsilon_1}]$ which agrees with the circuit $s_1(1^{(n)})$ on all n-bit inputs. Since $\epsilon_1 \leq b$, we have that $\mathbb{N} \not\models \psi_{a,b}(n, s_1(1^{(n)}), C_1^*)$.

We can apply a similar argument to the next disjunct using the aforementioned circuit C_1^*. In more detail, we consider the input $(x, \langle C_1^* \rangle)$ on D-Eval, which has length $m = n + 9n^{\epsilon_1}\log(n^{\epsilon_1}) + c \leq n^{\delta_2}$ for sufficiently large n due to $\delta_2 > \epsilon_1$, and a corresponding circuit $C_2^* \in \mathsf{CIRCUIT}[m^{1+\varepsilon_0}]$ provided by the circuit upper bound hypothesis. Similarly, we can fix the $9n^{\epsilon_1}\log(n^{\epsilon_1}) + c$ variables not related to the input string x. This provides an n-bit circuit $C_2^* \in \mathsf{CIRCUIT}[n^{\epsilon_2}]$ that computes according to the circuit $s_2(1^{(n)}, C_1^*)$, due to the definition of the D-Eval algorithm. Since $\epsilon_2 < b$, we have that $\mathbb{N} \not\models \psi_{a,b}(n, s_2(1^{(n)}, C_1^*), C_2^*)$.

Inductively, if we have circuits $C_1^*, C_2^*, \ldots, C_i^*$ for some $i \leq k-1$ of sizes at most $n^{\epsilon_1}, n^{\epsilon_2}, \ldots, n^{\epsilon_i}$, respectively, we consider the input $(x, \langle C_1^*, \ldots, C_i^* \rangle)$ to D-Eval, which has length $m = n + 9n^{\epsilon_1}\log(n^{\epsilon_1}) + \cdots + 9n^{\epsilon_i}\log(n^{\epsilon_i}) + c \leq n^{\delta_{i+1}}$ for sufficiently large n. Therefore, by taking a corresponding $m^{1+\varepsilon_0}$-size circuit for D-Eval and fixing all the inputs except for x, we get the circuit $C_{i+1}^* \in \mathsf{CIRCUIT}[n^{\epsilon_{i+1}}] \subseteq \mathsf{CIRCUIT}[n^b]$ which agrees with the circuit $s_{i+1}(1^{(n)}, C_1^*, \ldots, C_i^*)$ on all n-bit inputs. Consequently, $\mathbb{N} \not\models \psi_{a,b}(n, s_{i+1}(1^{(n)}, C_1^*, \ldots, C_i^*), C_{i+1}^*)$.

Overall, we can refute all disjuncts in Equation (1), which gives us a contradiction. This completes the proof. ◀

▶ **Theorem 7** (Theorem 1 Item (ii)). *If there are rationals $a > b > 1$ and $n_0 \in \mathbb{N}$ such that*

$$\mathsf{T}_2^1 \vdash \mathsf{CSH}[a, b, n_0]\,,$$

then there is a constant $\varepsilon > 0$ and a language $L \in \mathsf{P}^{\mathsf{NP}}$ such that $L \notin \mathsf{SIZE}[n^{1+\varepsilon}]$.

Proof. In this case, provability in T_2^1 provides by the KPT Theorem (Theorem 5) functions $s_1, \ldots, s_k$ which are in $\mathsf{FP}^{\mathsf{NP}}$ instead of FP as in the previous proof. Therefore, the algorithm D-Eval is in P^{NP} and we use the upper bound $\mathsf{P}^{\mathsf{NP}} \subseteq \bigcap_{\varepsilon>0} \mathsf{SIZE}[n^{1+\varepsilon}]$ to get a contradiction in the same way as above. ◀

Note that in the arguments above we have no control over the constant $\varepsilon > 0$. It depends on the number of disjuncts obtained from the KPT Theorem, which depends on the supposed proof of the hierarchy sentence.

3.2 Extracting All the Hardness from Proofs of a Succinct Hierarchy Theorem

In this section, we prove Theorem 2 Item (ii).

▶ **Theorem 8** (Theorem 2 Item (ii)). *If there are rationals $a > b > 1$ and a constant $n_0 \in \mathbb{N}$ such that*

$$\mathsf{T}^1_2 \vdash \mathsf{SCSH}[a, b, n_0]\,,$$

then there is a language $L \in \mathsf{P}^{\mathsf{NP}}$ such that $L \notin \mathsf{SIZE}[n^b]$.

Proof. The main idea here is to use the proof of SCSH in order to define a Turing machine M which runs in polynomial time using an NP oracle and its language is hard against n^b-size circuits.

Starting from $\mathsf{T}^1_2 \vdash \mathsf{SCSH}[a, b, n_0]$, we see that the structure of the sentence is $\forall\exists\forall\exists$:

$$\mathsf{SCSH}[a, b, n_0] \triangleq \forall n \geq n_0 \in \mathsf{Log},\ \exists \text{ collection } \mathcal{F},\ \forall \text{ circuit } C\ \ \phi_{a,b}(n, \mathcal{F}, C),$$

where $\phi_{a,b}(n, \mathcal{F}, C)$ is the formula that states that $\mathcal{F}$ is a collection $\{(x^1, b^1), \ldots, (x^\ell, b^\ell)\}$ with $\ell \leq n^a$, where $|x^i| = n$ and $|b^i| = 1$, and that if C is a circuit on n variables and of size $\leq n^b$, then there is some $i \in [\ell]$ such that $C(x^i) \neq b^i$ (we can move the existential quantifier at the front of the formula).

Thus, by the KPT Theorem (Theorem 5), there are $\mathsf{FP}^{\mathsf{NP}}$ functions $f_1, \ldots, f_k$, where k is a fixed constant, such that

$$\mathbb{N} \models \phi_{a,b}(n, f_1(1^{(n)}), C_1) \vee \phi_{a,b}(n, f_2(1^{(n)}, C_1), C_2) \vee \cdots \vee \phi_{a,b}(n, f_k(1^{(n)}, C_1, \ldots, C_{k-1}), C_k). \tag{2}$$

From the relation above, we can see that one of the functions $f_1, \ldots, f_k$ will output a collection that refutes every circuit of size $\leq n^b$. If it is not f_1, then there is a counterexample circuit C_1, which is used as extra input in f_2 and so on. Since $f_1, \ldots, f_k$ are in $\mathsf{FP}^{\mathsf{NP}}$, we can simulate this procedure in a P^{NP} Turing machine $M_{a,b}$, described below.

▶ Remark. In contrast with Algorithm 1, the algorithm of the Turing machine $M_{a,b}$ does not need to have the counterexample circuits as input, since it can guess and check them during its process, using the NP oracle. This difference in the input size is what gives us the n^b lower bound instead of $n^{1+\epsilon}$.

It is easy to see that the language $L(M_{a,b})$ recognised by the Turing machine $M_{a,b}$, is in P^{NP}. It suffices to show that $L(M_{a,b}) \notin \mathsf{SIZE}[n^b]$.

Consider a circuit $C \in \mathsf{CIRCUIT}[n^b]$. We will show that it fails to recognise $L(M_{a,b})$. Assume that the for-loop in Algorithm 2 ends in the r-th iteration with $r \leq k$. We fix the circuits $C_1, C_2, \ldots, C_{r-1}$ found by the algorithm. Then the formula $\phi_{a,b}(n, f_r(1^{(n)}, C_1, \ldots, C_{r-1}), C)$ always holds. If $r < k$ and C did not satisfy it, then the NP oracle would find C as a counterexample and it would continue to the $(r+1)$-th iteration. If $r = k$, then by the construction of $C_1, C_2, \ldots, C_{k-1}$, the formulas $\phi_{a,b}(n, f_i(1^{(n)}, C_1, \ldots, C_{i-1}), C_i)$ for $i < k$ do not hold, which means by Equation (2) that $\phi_{a,b}(n, f_k(1^{(n)}, C_1, \ldots, C_{k-1}), C)$ is true.

Algorithm 2 The Turing machine $M_{a,b}$, whose language is hard for n^b-size circuits.

Input : A bit-string x
Define $n \triangleq |x|$;
for $i = 1, \ldots, k$ **do**
  Simulate f_i with input $1^{(n)}$ and, if $i > 1$, $C_1, \ldots, C_{i-1}$. Interpret the output as a collection $\mathcal{F} = \{(x^1, b^1), \ldots, (x^\ell, b^\ell)\}$ with $\ell = n^a$;
  Check with an NP oracle whether there exists a circuit C of size $\leq n^b$, such that $C(x^i) = b^i$ for all $i \in [\ell]$;
  If not or if $i = k$, exit the for-loop with the current $\mathcal{F}$;
  If there is such a circuit, then use the NP oracle to find it and name it C_i.
end
If the pair $(x, 1)$ is in the collection $\mathcal{F}$, then **accept**. Else **reject**.

Since $\mathcal{F} \equiv f_r(1^{(n)}, C_1, \ldots, C_{r-1})$, from $\phi_{a,b}(n, \mathcal{F}, C)$ we get that there is some $i \in [\ell]$, such that $C(x^i) \neq b^i$. However, if $b^i = 1$, then $x^i \in L(M_{a,b})$, and if $b^i = 0$, then $x^i \notin L(M_{a,b})$. In both cases, the circuit C fails to recognise the language $L(M_{a,b})$, and the proof is complete. ◀

3.3 Formalization in T^2_2

In this section, we prove Theorem 1 Item (i) and Theorem 2 Item (i). To achieve this, we show that the succinct circuit size hierarchy is provable in APC_1, which is contained in T^2_2. We then observe that the circuit size hierarchy is easily provable from the succinct circuit size hierarchy.

▶ **Theorem 9.** *For every choice of rationals $a > b > 1$ and for every large enough $n_0 \in \mathbb{N}$,*

$$\mathsf{APC}_1 \vdash \mathsf{SCSH}[a, b, n_0]\,.$$

In particular, $\mathsf{SCSH}[a, b, n_0]$ *is provable in* T^2_2.

Proof. We define the polynomial-time function, f, which takes as input the description of a circuit, C, of size n^b, which means that the length of the description of C is $9n^b \log n^b$, and outputs a bit string y of length n^a with the property that for all $i = 0, 1, \ldots, n^a - 1$, $y_i = C(i)$.

The correctness of the polynomial-time algorithm f is provable in PV_1. In other words,

$$\mathsf{PV}_1 \vdash \forall n \in \mathsf{Log}\,(\,|x| \leq 9n^b \log n^b \,\wedge\, |y| \leq n^a\,) \rightarrow (\,|f(x)| \leq n^a \,\wedge\, (f(x) = y \leftrightarrow \forall i < n^a\; y_i = \mathsf{Eval}(x, i))). \tag{3}$$

The quantifier $\forall i \leq n^a$ is sharply bounded, so this formula is provable in PV_1.

The theory APC_1 includes the dWPHP axiom for all PV functions with input length n and output length $n + 1$, or equivalently input length n and output length m with $n < m$. From the first part of Equation (3), the input length of f is $9n^b \log n^b$, while the output length is n^a. Furthermore, it is provable in PV_1 that there is some constant n_0, such that $\forall n \geq n_0\; n^a > 9n^b \log n^b$. Therefore, we can use the axiom:

$$\mathsf{dWPHP}(f) \triangleq \forall n \geq n_0\; \exists y\; (|y| = n^a)\; \forall x\; (|x| = 9n^b \log n^b)\; f(x) \neq y \tag{4}$$

Every circuit of size n^b can be described by a string of size $9n^b \log n^b$, which means that

$$\forall C \in \mathsf{CIRCUIT}[n^b]\; |C| \leq 9n^b \log n^b.$$

Also, from the second part of Equation (3), using the notation for the circuit C, we get that

$$f(C) \neq y \leftrightarrow \exists i < n^a \; C(i) \neq y_i.$$

Substituting the last two relations to Equation (4), we get that

$$\mathsf{APC}_1 \vdash \forall n \geq n_0 \in \mathsf{Log} \; \exists y \; (|y| = n^a) \; \forall C \in \mathsf{CIRCUIT}[n^b] \; \exists i < n^a \; C(i) \neq y_i,$$

which is equivalent with $\mathsf{SCSH}[a, b, n_0]$, if we interpret y as the collection

$$\mathcal{F}_y \triangleq \{(0, y_0), (1, y_1), \ldots\}. \qquad \blacktriangleleft$$

▶ **Corollary 10.** *For every choice of rationals $a > b > 1$ and for every large enough $n_0 \in \mathbb{N}$,*

$$\mathsf{T}^2_2 \vdash \mathsf{CSH}[a, b, n_0]\,.$$

Proof. Since $a > b$, there is some rational $\epsilon > 0$, such that $a - \epsilon > b$. From Theorem 9, we have got a collection $\mathcal{F} = \{(x^1, b^1), \ldots, (x^\ell, b^\ell)\}$ of size $\ell \leq n^{a-\epsilon}$, such that for all circuits C of size less than n^b, there exists $i \in [\ell]$ such that $C(x^i) \neq b^i$. So, we only need to prove that

$$\mathsf{PV}_1 \vdash \exists \text{ circuit } D\colon \{0,1\}^n \to \{0,1\} \text{ of size} \leq n^a, \; \forall i \in [\ell] \; D(x^i) = b^i,$$

and then we can easily deduce that $\mathsf{APC}_1 \vdash \mathsf{CSH}[a, b, n_0]$. The same holds also for T^2_2.

It is sufficient to argue in PV_1 that there is a polynomial-time function $\mathsf{Circuit}(\mathcal{F})$ such that given the collection $\mathcal{F}$ from Theorem 9 outputs a circuit $D\colon \{0,1\}^n \to \{0,1\}$ of the required size such that $\forall i \in [\ell] \; D(x^i) = b^i$. In order to optimize the circuit size, we use that the obtained collection has a specific structure. More precisely, we have that for any $i \in [\ell]$, the strings x^i is the n-bit binary representation of the integer $i - 1$. Therefore, we can construct the circuit D in the following way: For every n-bit string x^i such that $(x^i, 1) \in \mathcal{F}$, we construct the term T^i, which is the conjunction of the first $|\ell|$ least significant bits of x^i (we put the literal z_j if the j-th bit of x^i is 1 and $\neg z_j$ if the j-th bit of x^i is 0, where $j \leq |\ell|$). Then we make the DNF

$$D \triangleq \bigvee_{(x^i,1)\in\mathcal{F}} T^i.$$

It is easy to see that D agrees with all the pairs of the collection $\mathcal{F}$. For an arbitrary pair (x^i, b^i), if $b^i = 1$, then the bits of x^i satisfy the term T^i, hence $D(x^i) = 1$. Otherwise, if $b^i = 0$, we know that the first $|\ell|$ least significant bits of x^i do not satisfy any term of the disjunction (since for all i, $x^i \leq \ell$), thus we get that $D(x^i) = 0$.

The DNF D can be viewed as a circuit and its correctness is easily provable in PV_1. This circuit has size at most $n^{a-\epsilon}|\ell|$ (derived by $|\ell| - 1$ $\wedge$-gates for each one of the at most $n^{a-\epsilon}$ terms and at most $n^{a-\epsilon}$ $\vee$-gates for the final disjunction), which is at most $n^{a-\epsilon}(\log n^{a-\epsilon} + 1)$. For large enough n_0, we can prove that $\forall n \geq n_0, \; n^{a-\epsilon}(\log n^{a-\epsilon} + 1) \leq n^a$, hence we have the desired result. $\blacktriangleleft$

4 Provability of Formula Size Bounds in PV_1

In this section, we prove Theorem 3. To achieve this, we establish that:

1. The parity function on n bits requires formulas of size $\geq n^{3/2}$ (Section 4.1).
2. The parity function on n bits can be computed by formulas of size $O(n^2) \leq n^a$ for any fixed rational $a > 2$ and large enough n (Section 4.2).
3. Consequently, the formula size hierarchy holds with parameters $a > 2$ and $b = 3/2$, provided that n_0 is large enough (Section 4.3).

4.1 Subbotovskaya's Lower Bound

4.1.1 High-Level Details of the Formalization

In this section, we sketch a formalization in PV_1 of the proof that the parity function on n bits requires Boolean formulas of size $\geq n^{3/2}$ [25].[3] We adapt the argument presented in [14, Section 6.3], which proceeds as follows:

1. [14, Lemma 6.8]: Given a Boolean formula F on n-bit inputs, it is possible to fix one of its variables so that the resulting formula F_1 satisfies

   $$\mathsf{Size}(F_1) \leq (1 - 1/n)^{3/2} \cdot \mathsf{Size}(F).$$

   In order to pick the variable to be restricted and its value, one first "normalizes" the formula F, as implicitly described in [14, Claim 6.9] (see more details below).
2. [14, Theorem 6.10]: By applying this result $\ell \triangleq n - k$ times, it is possible to obtain a formula F_ℓ on k-bit inputs such that

   $$\mathsf{Size}(F_\ell) \leq \mathsf{Size}(F) \cdot (1-1/n)^{3/2} \cdot (1-1/(n-1))^{3/2} \dots (1-1/(k+1))^{3/2} = \mathsf{Size}(F) \cdot (k/n)^{3/2}.$$

3. [14, Example 6.11]: If the initial formula F computes the parity function, by setting $\ell = n - 1$ we obtain

   $$1 \leq \mathsf{Size}(F_\ell) \leq (1/n)^{3/2} \cdot \mathsf{Size}(F),$$

   and consequently $\mathsf{Size}(F) \geq n^{3/2}$.

We recommend reading this section with [14, Section 6.3] at hand. We will slightly modify the argument when formalizing the lower bound in PV_1. In more detail, given a small formula F, we recursively construct (and establish correctness by induction) an n-bit input y witnessing that F does not compute the parity function. (Actually, for technical reasons related to the induction step, we will simultaneously construct an n-bit input y_n^0 witnessing that F does not compute the parity function and an n-bit input y_n^1 witnessing that F does not compute the negation of the parity function.)

Let $s(n)$ be a size bound and $\oplus(x)$ be a PV function that computes the parity of the binary string described by x, i.e., $\oplus(x) \triangleq x_1 \oplus x_2 \oplus \ldots \oplus x_n$, where x_i denotes the i-th bit of x. To simplify notation, we tacitly view x as a binary string. We assume that the formalization employs a well-behaved function symbol $\oplus$ such that PV_1 proves the basic properties of the parity function, e.g., $\mathsf{PV}_1 \vdash \oplus(x1) = 1 - \oplus(x)$ and $\mathsf{PV}_1 \vdash \oplus(x0) = \oplus(x)$.

We consider the following $\mathcal{L}_{\mathsf{PV}}$-sentence stating that the parity function requires formulas of size at least $s(n)$ for every input length $n \geq 1$:

$$\mathsf{FLB}_s \triangleq \forall N\, \forall n\, \forall F\, (n = |N| \geq 1 \wedge \mathsf{Size}(F) < s(n) \rightarrow \exists x\, (|x|_\ell = n \wedge \mathsf{Eval}(F, x) \neq \oplus(x)),$$ [4]

where for convenience of notation we use the function symbol $|w|_\ell$ to compute the bit-length of the string represented by w (under some reasonable encoding).

▶ **Theorem 11.** *Let* $s(n) \triangleq n^{3/2}$. *Then* $\mathsf{PV}_1 \vdash \mathsf{FLB}_s$.

[3] For concreteness, we let the size of a Boolean formula F be the number of leaves of F labeled by an input literal. We allow leaves that are labeled by constants, but we do not charge for them. Consequently, a constant function has formula complexity 0, while a non-constant function has formula complexity at least 1.

Proof. Given $b \in \{0,1\}$, we introduce the function $\oplus^b(x) \triangleq \oplus(x) + b \pmod 2$. In order to prove FLB_s in PV_1, we explicitly consider a polynomial-time function $R(1^{(n)}, F, b)$ with the following properties:[5]

1. Let $b \in \{0,1\}$.
2. If $\mathsf{Size}(F) < s(n)$ then $R(1^{(n)}, F, b)$ outputs an n-bit string y_n^b such that $\mathsf{Eval}(F, y_n^b) \neq \oplus^b(y_n^b)$.

In other words, $R(1^{(n)}, F, b)$ witnesses that the formula F does not compute the function $\oplus^b$ over n-bit strings. Note that the correctness of R is captured by the bounded universal sentence:

$$\mathsf{Ref}_{R,s} \triangleq \forall 1^{(n)}\, \forall F\, (\mathsf{Size}(F) < s(n) \to |y_n^0|_\ell = |y_n^1|_\ell = n \wedge F(y_n^0) \neq \oplus^0(y_n^0) \wedge F(y_n^1) \neq \oplus^1(y_n^1))\,,$$

where we employed the abbreviations $y_n^0 \triangleq R(1^{(n)}, F, 0)$ and $y_n^1 \triangleq R(1^{(n)}, F, 1)$. Our plan is to define R and show that $\mathsf{PV}_1 \vdash \mathsf{Ref}_{R,s}$. Note that this implies FLB_s in PV_1. Jumping ahead, the correctness of $R(1^{(n)}, F, b)$ will be established by polynomial induction on N (equivalently, induction on $n = |N|$). Since $\mathsf{Ref}_{R,s}$ is a universal sentence and S_2^1 is $\forall\Sigma_1^b$-conservative over PV_1, polynomial induction for NP and coNP predicates (admissible in S_2^1; see, e.g., [16, Section 5.2]) is available during the formalization. More details follow.

The procedure $R(1^{(n)}, F, b)$ makes use of a few polynomial-time sub-routines (discussed below) and is defined in the following way:

Algorithm 3 Refuter Algorithm $R(1^{(n)}, F, b)$.

Input : $1^{(n)}$ for some $n \geq 1$, formula F over n-bit inputs, $b \in \{0,1\}$.

1 Let $s(n) \triangleq n^{3/2}$. If $\mathsf{Size}(F) \geq s(n)$ **return** *"error"*;

2 If $\mathsf{Size}(F) = 0$, F computes a constant function $b_F \in \{0,1\}$. In this case, **return** *the n-bit string $y_n^b \triangleq y_1^b 0^{n-1}$ such that $\oplus^b(y_1^b 0^{n-1}) \neq b_F$*;

3 Let $\widetilde{F} \triangleq \mathsf{Normalize}(1^{(n)}, F)$;

// $\widetilde{F}$ satisfies [14, Claim 6.9], $\mathsf{Size}(\widetilde{F}) \leq \mathsf{Size}(F)$, $\forall x \in \{0,1\}^n\ F(x) = \widetilde{F}(x)$.

4 Let $\rho \triangleq \mathsf{Find\text{-}Restriction}(1^{(n)}, \widetilde{F})$, where $\rho\colon [n] \to \{0,1,\star\}$ and $|\rho^{-1}(\star)| = n - 1$;

// ρ restricts a suitable variable x_i to a bit c_i, as in [14, Lemma 6.8].

5 Let $F' \triangleq \mathsf{Apply\text{-}Restriction}(1^{(n)}, \widetilde{F}, \rho)$. Moreover, let $b' \triangleq b \oplus c_i$ and $n' \triangleq n - 1$;

// F' is an n'-bit formula; $\forall z \in \{0,1\}^{\rho^{-1}(\star)}\ F'(z) = \widetilde{F}(z \cup x_i \mapsto c_i)$.

6 Let $y_{n'}^{b'} \triangleq R(1^{n'}, F', b')$ and **return** *the n-bit string $y_n^b \triangleq y_{n'}^{b'} \cup y_i \mapsto c_i$*;

4.1.1.1 Normalize$(1^{(n)}, F)$ and its properties (in S_2^1)

We say that a subformula G of F is a *neighbor* of a leaf z if either $z \wedge G$ or $z \vee G$ is a subformula of F. We say that a formula F over variables $\{x_1, \ldots, x_n\}$ is in *normal form* if for every $i \in [n]$ and every literal $z \in \{x_i, \overline{x_i}\}$, if z is a leaf of F and G is a neighbor of z in F, then G does not contain the variable x_i.

[5] For convenience, we often write $1^{(n)}$ instead of explicitly considering parameters N and $n = |N|$. We might also write just $F(x)$ instead of $\mathsf{Eval}(F, x)$.

▶ **Lemma 12.** *There is a polynomial-time function* $\mathsf{Normalize}(1^{(n)}, F)$ *that given a Boolean formula* F *over* n *input variables, outputs a formula* $\widetilde{F}$ *over* n *input variables such that the following holds:*

(i) $\mathsf{Size}(\widetilde{F}) \leq \mathsf{Size}(F)$.
(ii) *For every input* $x \in \{0,1\}^n$, $\widetilde{F}(x) = F(x)$.
(iii) $\widetilde{F}$ *is in normal form.*
(iv) $\widetilde{F}$ *is either a constant* 0 *or* 1*, or* $\widetilde{F}$ *contains no leaves labeled by constants* 0 *and* 1.

Moreover, the correctness of $\mathsf{Normalize}(1^{(n)}, F)$ *is provable in* S^1_2.

Proof Sketch. It is enough to verify that the proof of [14, Claim 6.9] provides such a polynomial-time function and that its correctness can be established in S^1_2. In more detail, if F is not in normal form, we can efficiently compute a literal $z \in \{x_i, \overline{x_i}\}$ and a neighbor G of z that violates the corresponding property. As shown in [14, Claim 6.9], we can fix any leaf $z' \in \{x_i, \overline{x_i}\}$ in G by an appropriate constant c so that the resulting formula F_1 satisfies conditions (i) and (ii) of Lemma 12. After at most $\ell \triangleq \mathsf{Size}(F)$ iterations, we obtain a sequence $F_1, \ldots, F_\ell$ of formulas such that $\widetilde{F} \triangleq F_\ell$ satisfies conditions (i), (ii), and (iii) of the lemma. Moreover, condition (iv) can always be guaranteed by simplifying the final formula, i.e., by replacing subformulas $0 \vee G$ by G, $1 \vee G$ by 1, $0 \wedge G$ by 0, and $1 \wedge G$ by G. The correctness of $\widetilde{F} \triangleq \mathsf{Normalize}(1^{(n)}, F)$ can be established by polynomial induction for coNP predicates (i.e., Π^b_1 formulas), which is available in S^1_2. ◀

4.1.1.2 Find-Restriction$(1^{(n)}, \widetilde{F})$ and its properties (in S^1_2)

We argue in S^1_2 and follow the argument from the proof of [14, Lemma 6.8]. Let $\widetilde{F}$ be a formula over n input variables in normal form. We focus on the non-trivial case, and assume that $n \geq 2$, $\mathsf{Size}(\widetilde{F}) \geq 2$, and that $\widetilde{F}$ contains no leaves labeled by constants. Let $\mathsf{Count}(1^{(n)}, F, i)$ be a polynomial-time algorithm that outputs the number of leaves of F that contain the variable x_i (including its appearances as $\overline{x_i}$). Let $w = (w_1, \ldots, w_n)$ be the corresponding sequence of multiplicities, i.e., $w_i \triangleq \mathsf{Count}(1^{(n)}, F, i)$. Note that $\sum_i w_i = \widetilde{s}$, where $\widetilde{s} \triangleq \mathsf{Size}(\widetilde{F})$.

We claim that S^1_2 proves the existence of an index $i \in [n]$ such that $w_i \geq \widetilde{s}/n$. First, for each $j \in [n]$, we define the cumulative sum $v_j \triangleq \sum_{i \leq j} w_j$. Let $v \triangleq (v_0, v_1, \ldots, v_n)$ be the corresponding sequence, where we set $v_0 \triangleq 0$. Notice that $v_n = \widetilde{s}$. Since v contains $n+1$ elements, it can be efficiently computable from w. We now argue by induction on n that for some index $j \in [n]$ we have $v_j - v_{j-1} \geq v_n/n$. This implies that $w_j = v_j - v_{j-1} \geq v_n/n = \widetilde{s}/n$, as desired.

If $n = 1$, then $v_1 - v_0 = v_1 = v_1/1$ and the result holds for $j = 1$. Assume the result holds for $n-1$, and consider v_n. If $v_n - v_{n-1} \geq v_n/n$, we can pick $j = n$ and we are done. Otherwise, $v_{n-1} \geq v_n - v_n/n = v_n(n-1)/n$. By the induction hypothesis, there is an index $j \in [n-1]$ such that $v_j - v_{j-1} \geq v_{n-1}/(n-1)$. Using the lower bound on v_{n-1}, we get that $v_j - v_{j-1} \geq v_n/n$, which concludes the proof.

Consequently, S^1_2 proves the existence of a variable x_i which appears $t \geq \widetilde{s}/n$ times as a leaf of $\widetilde{F}$. Let $z_1, \ldots, z_t$ be the leaves of $\widetilde{F}$ labeled by either x_i or $\overline{x_i}$. Recall that we assume that $n \geq 2$, $\mathsf{Size}(\widetilde{F}) \geq 2$, and that $\widetilde{F}$ satisfies conditions (iii) and (iv) of Lemma 12. Therefore, each leaf z_j has a neighbor subformula G_j in $\widetilde{F}$ that contains some leaf labeled by a literal not in $\{x_i, \overline{x_i}\}$. For this reason, if we set x_i to an appropriate constant c_j, G_j will disappear from F, thereby erasing at least another leaf not among $z_1, \ldots, z_t$. As in the proof of [14, Lemma 6.8], if we let $c \in \{0,1\}$ be the constant that appears more often among

$c_1, \ldots, c_t$ and set $x_i \mapsto c$ in the restriction ρ, all the leaves $z_1, \ldots, z_t$ will be eliminated from $\widetilde{F}$ together with at least $t/2$ additional leaves.[6] Thus the total number of eliminated leaves, which we specify using a polynomial-time function $\mathsf{NumRemoved}(1^{(n)}, \widetilde{F}, \rho)$, satisfies

$$\mathsf{NumRemoved}(1^{(n)}, \widetilde{F}, \rho) \geq t + \frac{t}{2} \geq \frac{3\widetilde{s}}{2n}.$$

Overall, it follows that

$$\mathsf{S}^1_2 \vdash \widetilde{F} = \mathsf{Normalize}(1^{(n)}, F) \wedge \rho = \mathsf{Find\text{-}Restriction}(1^{(n)}, \widetilde{F}) \rightarrow \mathsf{NumRemoved}(1^{(n)}, \widetilde{F}, \rho) \geq \frac{3}{2n} \cdot \mathsf{Size}(\widetilde{F})\,.$$

4.1.1.3 Apply-Restriction$(1^{(n)}, \widetilde{F}, \rho)$ and its properties (in S^1_2)

We only sketch the details. This is simply a polynomial-time algorithm that, given a formula $\widetilde{F}$ on n input variables and a restriction $\rho\colon [n] \to \{0, 1, *\}$ with $|\rho^{-1}(\star)| = n - 1$ (i.e., ρ restricts a single variable x_i to a constant $c_i \in \{0, 1\}$), outputs a formula F' over $n-1$ input variables that sets every literal $z \in \{x_i, \overline{x_i}\}$ to the corresponding constant and simplifies the resulting formula, e.g., replaces subformulas $0 \vee G$ by G, $1 \vee G$ by 1, $0 \wedge G$ by 0, and $1 \wedge G$ by G. Additionally, for $F' = \mathsf{Apply\text{-}Restriction}(1^{(n)}, \widetilde{F}, \rho)$, we have

$$\mathsf{S}^1_2 \vdash \mathsf{Size}(F') \leq \mathsf{Size}(\widetilde{F}) - \mathsf{NumRemoved}(1^{(n)}, \widetilde{F}, \rho) \wedge \forall z \in \{0,1\}^{\rho^{-1}(\star)}\ F'(z) = \widetilde{F}(z \cup x_i \mapsto c_i)\,. \tag{5}$$

Using the computed bound on $\mathsf{NumRemoved}(1^{(n)}, \widetilde{F}, \rho)$ for $\rho = \mathsf{Find\text{-}Restriction}(1^{(n)}, \widetilde{F})$, we obtain that for $\widetilde{F}$ and F' defined as above (with $s' \triangleq \mathsf{Size}(F')$ and $\widetilde{s} \triangleq \mathsf{Size}(\widetilde{F})$), and assuming that $n \geq 2$,

$$\mathsf{S}^1_2 \vdash s' \leq \widetilde{s} - \frac{3}{2n} \cdot \widetilde{s} = \widetilde{s} \cdot \left(1 - \frac{3}{2n}\right) \leq \widetilde{s} \cdot \left(1 - \frac{1}{n}\right)^{3/2}. \tag{6}$$

The last inequality uses that $\mathsf{S}^1_2 \vdash \forall a,\ a \geq 2 \rightarrow (1 - 3/(2a))^2 \leq (1 - 1/a)^3$, which one can easily verify.

Note that $R(1^{(n)}, F, b)$ runs in time polynomial in $n + |F| + |b|$ and that it is definable in S^1_2. Next, we establish the correctness of $R(1^{(n)}, F, b)$ in S^1_2 .

▶ **Lemma 13.** *Let $s(n) \triangleq n^{3/2}$. Then $\mathsf{S}^1_2 \vdash \mathsf{Ref}_{R,s}$.*

Proof. We consider the formula $\varphi(N)$ defined as

$$\forall F\, \forall n = |N| \geq 1 (\mathsf{Size}(F) < s(n)) \rightarrow (|y^0_n|_\ell = |y^1_n|_\ell = n \wedge F(y^0_n) \neq \oplus^0(y^0_n) \wedge F(y^1_n) \neq \oplus^1(y^1_n))\,,$$

where as before we use $y^0_n \triangleq R(1^{(n)}, F, 0)$ and $y^1_n \triangleq R(1^{(n)}, F, 1)$. Note that $\varphi(N)$ is a Π^b_1 formula. Below, we argue that

$$\mathsf{S}^1_2 \vdash \varphi(1) \quad \text{and} \quad \mathsf{S}^1_2 \vdash \forall N\, \varphi(\lfloor N/2 \rfloor) \rightarrow \varphi(N)\,.$$

Then, by polynomial induction for Π^b_1 formulas (available in S^1_2) and using that $\varphi(0)$ trivially holds, it follows that $\mathsf{S}^1_2 \vdash \forall N\, \varphi(N)$. In turn, this yields $\mathsf{S}^1_2 \vdash \mathsf{Ref}_{R,s}$.

[6] The existence of such a constant c can be proved in S^1_2 in a way that is similar to the proof that some variable x_i appears in at least $\widetilde{s}/n$ leaves.

Base Case: $\mathsf{S}^1_2 \vdash \varphi(1)$. In this case, for a given formula F and length n, the hypothesis of $\varphi(1)$ is satisfied only if $n = 1$ and $\mathsf{Size}(F) = 0$. Let $y^0_1 \triangleq R(1, F, 0)$ and $y^1_1 \triangleq R(1, F, 1)$. We need to prove that

$$|y^0_1|_\ell = |y^1_1|_\ell = 1 \wedge F(y^0_1) \neq \oplus^0(y^0_1) \wedge F(y^1_1) \neq \oplus^1(y^1_1)\,.$$

Since $n = 1$ and $\mathsf{Size}(F) = 0$, F evaluates to a constant b_F on every input bit. The statement above is implied by Line 2 in the definition of $R(n, F, b)$.

(Polynomial) Induction Step: $\mathsf{S}^1_2 \vdash \forall N\, \varphi(\lfloor N/2 \rfloor) \to \varphi(N)$. Fix an arbitrary N, let $n \triangleq |N|$, and assume that $\varphi(\lfloor N/2 \rfloor)$ holds. By the induction hypothesis, for every formula F' with $\mathsf{Size}(F') < n'^{3/2}$, where $n' \triangleq n - 1$, we have

$$|y^0_{n'}|_\ell = |y^1_{n'}|_\ell = n' \;\wedge\; F'(y^0_{n'}) \neq \oplus^0(y^0_{n'}) \;\wedge\; F'(y^1_{n'}) \neq \oplus^1(y^1_{n'})\,, \tag{7}$$

where $y^0_{n'} \triangleq R(1^{n'}, F', 0)$ and $y^1_{n'} \triangleq R(1^{n'}, F', 1)$.

Now let $n \geq 2$, and let F be a formula over n-bit inputs of size $< n^{3/2}$. By the size bound on F, $R(1^{(n)}, F, b)$ ignores Line 1. If $\mathsf{Size}(F) = 0$, then similarly to the base case it is trivial to check that the conclusion of $\varphi(N)$ holds. Therefore, we assume that $\mathsf{Size}(F) \geq 1$ and $R(1^{(n)}, F, b)$ does not stop at Line 2. Let $\widetilde{F} \triangleq \mathsf{Normalize}(1^{(n)}, F)$ (Line 3), $\rho \triangleq \mathsf{Find\text{-}Restriction}(1^{(n)}, \widetilde{F})$ (Line 4), $F' \triangleq \mathsf{Apply\text{-}Restriction}(1^{(n)}, \widetilde{F}, \rho)$ (Line 5), $n' \triangleq n - 1$ (Line 5), and $b' \triangleq b \oplus c_i$ (Line 5), where ρ restricts the variable x_i to the bit c_i. Moreover, for convenience, let $s \triangleq \mathsf{Size}(F)$, $\widetilde{s} \triangleq \mathsf{Size}(\widetilde{F})$, and $s' \triangleq \mathsf{Size}(F')$. By Lemma 12 Item (i), Equation (6), and the bound $s < n^{3/2}$,

$$\mathsf{S}^1_2 \vdash s' \leq \widetilde{s} \cdot (1 - 1/n)^{3/2} \leq s \cdot (1 - 1/n)^{3/2} < n^{3/2} \cdot (1 - 1/n)^{3/2} = (n - 1)^{3/2}\,.$$

Thus F' is a formula on n'-bit inputs of size $< n'^{3/2}$. Recall that for a given $b \in \{0, 1\}$ we have $b' = b \oplus c_i$. Let $y^{b'}_{n'} \triangleq R(1^{n'}, F', b')$ (Line 6). By the first condition in the induction hypothesis (Equation (7)) and the definition of each $y^b_n \triangleq y^{b'}_{n'} \cup y_i \mapsto c_i$, we have $|y^0_n|_\ell = |y^1_n|_\ell = n$. Below, we also rely on the last two conditions in the induction hypothesis (Equation (7)), Lemma 12 Item (ii), and the last condition in Equation (5). We derive the following statements, where $b \in \{0, 1\}$:

$$\begin{aligned} F'(y^{b'}_{n'}) &\neq \oplus^{b'}(y^{b'}_{n'})\,,\\ F(y^b_n) &= F'(y^{b'}_{n'})\,,\\ F(y^b_n) &\neq \oplus^{b'}(y^{b'}_{n'})\,. \end{aligned}$$

Notice that

$$\oplus^{b'}(y^{b'}_{n'}) = \oplus^{b \oplus c_i}(y^{b'}_{n'}) = c_i \oplus (\oplus^b(y^{b'}_{n'})) = c_i \oplus (\oplus^b(y^b_n) \oplus c_i) = \oplus^b(y^b_n)\,.$$

These statements imply that, for each $b \in \{0, 1\}$, $F(y^b_n) \neq \oplus^b(y^b_n)$. In other words, the conclusion of $\varphi(N)$ holds. This completes the proof of the induction step. ◀

As explained above, the provability of $\mathsf{Ref}_{R,s}$ in S^1_2 implies its provability in PV_1. Since $\mathsf{PV}_1 \vdash \mathsf{Ref}_{R,s} \to \mathsf{FLB}_s$, this completes the proof of Theorem 11. ◀

4.1.2 On the Low-Level Details of the Formalization

In order to make our presentation accessible to a broader audience, in this section we provide more details about the formalization of algorithms and about the proofs of their basic properties. However, due to space restriction, the section is included only in the full version.

4.2 Upper Bound

In this section, we show that the parity function on n bits can be computed by formulas of size $O(n^2)$, provably in PV_1. We can formalize this upper bound in the language of PV, defining an $\mathcal{L}_{\mathsf{PV}}$-sentence stating that the parity function can be computed by a formula of size $s(n)$ for every input length $n \geq 1$:

$$\mathsf{FUB}_s \triangleq \forall N\, \forall n\, \exists F\, (n = |N| \geq 1 \wedge \mathsf{Size}(F) < s(n) \wedge \forall x\, (|x| \leq n \to \mathsf{Eval}(F, x) = \oplus_n^0(x))\,.$$

▶ **Theorem 14.** *Let $s(n) \triangleq 4n^2$. Then $\mathsf{PV}_1 \vdash \mathsf{FUB}_s$.*

Proof. FUB_s is a $\forall\Sigma_2^b$ sentence and our intended theory is PV_1. In order to implement some inductive proofs, it will be helpful to reduce the complexity of the formula. For this, we introduce a new polynomial-time function, $\mathsf{ParForm}(1^{(n)})$, which generates the desired formula that computes the parity function on n bits. Since it is a polynomial-time function, there is a symbol for it in PV and we can use it in the new formalization:

$$\begin{aligned}\mathsf{FUB}'_s \triangleq &\forall N\, \forall n\, (n = |N| \geq 1 \wedge \mathsf{Size}(\mathsf{ParForm}(1^{(n)})) < s(n) \wedge \\ &\forall x\, (|x| \leq n \to \mathsf{Eval}(\mathsf{ParForm}(1^{(n)}), x) = \oplus_n^0(x))\,.\end{aligned}$$

It is immediate that $\mathsf{FUB}'_s \Rightarrow \mathsf{FUB}_s$, thus we focus on proving FUB'_s. We continue with the following steps:

1. We prove an upper bound of n^2 for the formulas calculating the parity function and its negation, when n is a power of 2.
2. We use this construction to derive the $4n^2$ upper bound for any n.

Next, we define a polynomial-time algorithm $\mathsf{Par}(1^{(n)})$ which computes a formula that calculates the parity function on n bits and a formula that calculates the negation of the parity function on n bits, if n is a power of 2.

Algorithm 4 $\mathsf{Par}(1^{(n)})$ outputs Boolean formulas for $\oplus_n^0$ and $\oplus_n^1$ when n is a power of 2.

Input : $1^{(n)}$ for some $n \geq 1$.
Let $k \triangleq |n-1|$. If $n \neq 2^k$ (n is not a power of 2), then **return** *"error"*;
`// F will compute the parity function, while` $\overline{F}$ `will compute its negation`
if $k = 0$ **then**
  Define F to be the formula with one leaf x_1 and $\overline{F}$ to be the formula with one leaf $\neg x_1$.
else if $k \geq 1$ **then**
  `// Construct a pair` $(F, \overline{F})$ `of formulas on input bits` $x_1, \ldots, x_{2^k}$ `as follows:`
  Let $(F_1, \overline{F_1}) \triangleq \mathsf{Par}(1^{n/2})$, and define a corresponding pair $(F_2, \overline{F_2})$:
  In F_2 and $\overline{F}_2$, relabel the leaves by putting $x_{2^{k-1}+i}$ instead of x_i for every $i = 1, \ldots, 2^{k-1}$;
  Now let $F \triangleq (F_1 \vee F_2) \wedge (\overline{F}_1 \vee \overline{F}_2)$ and $\overline{F} \triangleq (F_1 \wedge F_2) \vee (\overline{F}_1 \wedge \overline{F}_2)$.
end
return $(F, \overline{F})$.

▶ **Lemma 15.** *If n is a power of 2, the algorithm $\mathsf{Par}(1^{(n)})$ correctly outputs two formulas $(F, \overline{F})$ of size n^2 which calculate the parity function and its negation, provably in $\mathsf{S}_2^1(\mathsf{PV})$.*

Proof. We split the proof of the correctness for the algorithm $\mathsf{Par}(1^{(n)})$ into 3 properties:

1. $\phi_1(n) \triangleq F, \overline{F} \in \mathsf{VALIDFORM}(n)$, where $\mathsf{VALIDFORM}(n)$ is the set of formulas on n variables;
2. $\phi_2(n) \triangleq \mathsf{Size}(F) = \mathsf{Size}(\overline{F}) = n^2$;
3. $\phi_3(n) \triangleq \forall x\ |x| \leq n \to \mathsf{Eval}(F, x) = \oplus_n^0(x) \wedge \mathsf{Eval}(\overline{F}, x) = \oplus_n^1(x)$.

For now we only care about the case that n is a power of 2, so we prove these properties conditionally (equivalently we prove $(n = (n-1)\#1) \to \phi(n)$).[7] That is why it suffices to use polynomial induction on n, which is available in S_2^1, since our formulas are at most Π_1^b.

We skip the proof of ϕ_1, which is proven by simple induction as below, using the fact that if F_1, F_2 are formulas then $F_1 \wedge F_2$ and $F_1 \vee F_2$ are also formulas.

Property 2: $\mathsf{S}_2^1 \vdash \phi_2(n)$. For the base case, $\phi_2(1)$, we have $k = 0$, which means that the output $(F, \overline{F}) \triangleq \mathsf{Par}(1^1)$ will be two formulas with one leaf each, hence

$$\mathsf{Size}(F) = \mathsf{Size}(\overline{F}) = 1.$$

For the induction step, we need $\mathsf{S}_2^1 \vdash \forall n\, \phi_2(\lfloor n/2 \rfloor) \to \phi_2(n)$. If n is not a power of 2, then the statement is true by default. In the case of n being a power of 2, we fix $k = |n-1|$ and we want to prove equivalently:

$$\mathsf{S}_2^1 \vdash \phi_2(2^{k-1}) \to \phi_2(2^k).$$

Assume that $\phi_2(2^{k-1}) \equiv \phi_2(n/2)$ holds. From Line 8 we have that

$$F = (F_1 \vee F_2) \wedge (\overline{F}_1 \vee \overline{F}_2) \text{ and } \overline{F} = (F_1 \wedge F_2) \vee (\overline{F}_1 \wedge \overline{F}_2), \tag{8}$$

where $(F_1, \overline{F_1})$ and $(F_2, \overline{F_2})$ are copies of $\mathsf{Par}(1^{n/2})$. From the induction hypothesis, this means that $\mathsf{Size}(F_1) = \mathsf{Size}(\overline{F_1}) = \mathsf{Size}(F_2) = \mathsf{Size}(\overline{F_2}) = (n/2)^2 = 2^{2(k-1)}$. Therefore, from (Equation (8)) and the properties of the function Size, we get

$$\mathsf{Size}(F) = \mathsf{Size}(F_1) + \mathsf{Size}(\overline{F_1}) + \mathsf{Size}(F_2) + \mathsf{Size}(\overline{F_2}) = 4 \cdot 2^{2(k-1)} = 2^{2k} = n^2.$$

Similarly for $\overline{F}$, which means that $\phi_2(2^k) \equiv \phi_2(n)$ holds. This completes the proof of the induction for ϕ_2.

Property 3: $\mathsf{S}_2^1 \vdash \phi_3(n)$. Here the base case is trivial: for $F \triangleq x_1$ and $x \in \{0, 1\}$, then $\mathsf{Eval}(F, x) = x = \oplus_1^0(x)$. Similarly for $\overline{F}$.

For the induction step, we assume as above that $n = 2^k$ and we want to prove:

$$\mathsf{S}_2^1 \vdash \phi_3(2^{k-1}) \to \phi_3(2^k).$$

We assume that $\phi_2(2^{k-1}) \equiv \phi_2(n/2)$ holds and we write F in the form

$$F = (F_1 \vee F_2) \wedge (\overline{F}_1 \vee \overline{F}_2) \text{ and } \overline{F} = (F_1 \wedge F_2) \vee (\overline{F}_1 \wedge \overline{F}_2),$$

where $(F_1, \overline{F_1})$ and $(F_2, \overline{F_2})$ are copies of $\mathsf{Par}(1^{n/2})$. Therefore, instead of $\mathsf{Eval}(F, x)$, we can calculate

$$\mathsf{Eval}((F_1 \vee F_2) \wedge (\overline{F}_1 \vee \overline{F}_2), x).$$

[7] It is easy to check that this is true if and only if n is a power of 2.

We need to prove that $\mathsf{Eval}(F, x) = \oplus^0_n(x)$ for all x with $|x| \leq n$. So, taking one such x we can split its binary representation into two parts x_1, x_2 with lengths $|x_1|, |x_2| \leq n/2$, such that $x = (x_2 x_1)_b = x_1 + 2^{n/2} x_2$.

The input to subformulas $F_2, \overline{F_2}$ from the definition are the bits x_{2^k-1+i} for $i = 1, \ldots, 2^{k-1}$, which means that their input is x_2. Similarly, the input to subformulas $F_1, \overline{F_1}$ is x_1. Hence, we can define

$$b_1 \triangleq \mathsf{Eval}(F_1, x_1) b_3 \triangleq \mathsf{Eval}(\overline{F_1}, x_1)$$
$$b_2 \triangleq \mathsf{Eval}(F_2, x_2) b_4 \triangleq \mathsf{Eval}(\overline{F_2}, x_2)$$

From the properties of the evaluation function and the form of F, we can prove in S^1_2 that $\mathsf{Eval}(F, x) = (b_1 \vee b_2) \wedge (b_3 \vee b_4)$, where the symbols $\vee, \wedge$ are used as Boolean symbols here.

However, since $|x_1|, |x_2| \leq n/2$ and $(F_1, \overline{F_1}) = (F_2, \overline{F_2}) = \mathsf{Par}(1^{n/2})$, from the induction hypothesis we get that

$$b_1 = \oplus^0(x_1) b_3 = \oplus^1(x_1) = 1 - b_1$$
$$b_2 = \oplus^0(x_2) b_4 = \oplus^1(x_2) = 1 - b_2$$

Next, it is easy to prove by checking all the 4 cases that

$$\forall b_1, b_2 \in \{0, 1\}\ (b_1 \vee b_2) \wedge ((1 - b_1) \vee (1 - b_2)) = b_1 \oplus b_2,$$

and as a result, we get

$$\mathsf{Eval}(F, x) = (\oplus^0(x_1)) \oplus (\oplus^0(x_2)) = \oplus^0(x_2 x_1) = \oplus^0(x)$$

by the properties of the parity function. Similarly, we can prove that $\mathsf{Eval}(\overline{F}, x) = \oplus^1_n(x)$, which concludes the induction. ◀

For the general case, we use a simple padding argument. For a number n, we can define the number

$$\tilde{n} \triangleq (n - 1)\#1.$$

This number is the least power of 2 that is greater or equal to n. It is easy to see that

$$\mathsf{PV}_1 \vdash n \leq \tilde{n} < 2n.$$

If we replace $\mathsf{ParForm}(1^{(n)})$ by $\mathsf{Par}_1(1^{\tilde{n}})$ (the first coordinate of $\mathsf{Par}(1^{\tilde{n}})$), we have by the above lemma that

1. $\mathsf{Size}(\mathsf{ParForm}(1^{(n)})) = \mathsf{Size}(\mathsf{Par}_1(1^{\tilde{n}})) = \tilde{n}^2 < (2n)^2 = s(n)$.
2. For all x with $|x| \leq n$, we have $|x| \leq \tilde{n}$, which by the lemma gives us

   $$\mathsf{Eval}(\mathsf{ParForm}(1^n), x) = \mathsf{Eval}(\mathsf{Par}_1(1^{\tilde{n}}), x) = \oplus^0_{\tilde{n}}(x).$$

   Since $|x| \leq n$, we also have $\oplus^0_{\tilde{n}}(x) = \oplus^0_n(x)$. Consequently, we have $\mathsf{Eval}(\mathsf{ParForm}(1^n), x) = \oplus^0_n(x)$.

These two together show that $\mathsf{PV}_1 \vdash \mathsf{FUB}'_s$ and the proof is complete. ◀

4.3 Formula Size Hierarchy

In this section, we provide the proof of Theorem 3.

▶ **Theorem 16** (Theorem 3). *Consider rationals $a > 2$ and $b = 3/2$, and let n_0 be a large enough positive integer. Then*

$$\mathsf{PV}_1 \vdash \mathsf{FSH}[a, b, n_0]\,.$$

Proof. We combine the results of Section 4.1 and Section 4.2. We argue in PV_1. From Theorem 11, we get that

$$\forall n \in \mathsf{Log}\ \forall F \in \mathsf{FORMULA}[n^{3/2}]\ \exists x\ (|x| \leq n \wedge F(x) \neq \oplus_n(x)), \tag{9}$$

and from Theorem 14, we have that

$$\forall n \in \mathsf{Log}\ \exists G \in \mathsf{FORMULA}[4n^2]\ \forall x\ (|x| \leq n \to G(x) = \oplus_n(x)).$$

We can eliminate the constant 4 from the latter using that $a > 2$ and choosing a large enough n_0, such that for every $n \geq n_0$, $n^a \geq 4n^2$ (provably in PV_1). Consequently,

$$\forall n \geq n_0 \in \mathsf{Log}\ \exists G \in \mathsf{FORMULA}[n^a]\ \forall x\ (|x| \leq n \to G(x) = \oplus_n(x)). \tag{10}$$

Finally, combining Equation (9) and Equation (10), we get that

$$\forall n \geq n_0 \in \mathsf{Log}\ \exists G \in \mathsf{FORMULA}[n^a]\ \forall F \in \mathsf{FORMULA}[n^{3/2}]\ \exists x\ (|x| \leq n\ \wedge\ F(x) \neq G(x)),$$

which is exactly the formula size hierarchy, $\mathsf{FSH}[a, b, n_0]$, for our choice of parameters $a > 2$ and $b = 3/2$. ◀

References

1 Sanjeev Arora and Boaz Barak. *Computational Complexity – A Modern Approach*. Cambridge University Press, 2009. URL: http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264.

2 Samuel R. Buss. *Bounded Arithmetic*. Bibliopolis, 1986.

3 Samuel R. Buss. Bounded arithmetic and propositional proof complexity. In *Logic of Computation*, pages 67–121. Springer Berlin Heidelberg, 1997.

4 Lijie Chen, Jiatu Li, and Igor C. Oliveira. Reverse mathematics of complexity lower bounds. In *Symposium on Foundations of Computer Science* (FOCS), 2024.

5 Lijie Chen, Dylan M. McKay, Cody D. Murray, and R. Ryan Williams. Relations and equivalences between circuit lower bounds and Karp-Lipton theorems. In *Computational Complexity Conference* (CCC), pages 30:1–30:21, 2019. doi:10.4230/LIPIcs.CCC.2019.30.

6 Alan Cobham. The intrinsic computational difficulty of functions. *Proc. Logic, Methodology and Philosophy of Science*, pages 24–30, 1965.

7 Stephen A. Cook. Feasibly constructive proofs and the propositional calculus (preliminary version). In *Symposium on Theory of Computing* (STOC), pages 83–97, 1975. doi:10.1145/800116.803756.

8 Stephen A. Cook and Phuong Nguyen. *Logical Foundations of Proof Complexity*. Cambridge University Press, 2010. doi:10.1017/CBO9780511676277.

9 Russell Impagliazzo and Avi Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *Symposium on the Theory of Computing* (STOC), pages 220–229, 1997. doi:10.1145/258533.258590.

10 Emil Jeřábek. Dual weak pigeonhole principle, boolean complexity, and derandomization. *Annals of Pure and Applied Logic*, 129(1-3):1–37, 2004. doi:10.1016/j.apal.2003.12.003.

11 Emil Jeřábek. *Weak pigeonhole principle and randomized computation*. PhD thesis, Charles University in Prague, 2005.

12 Emil Jeřábek. The strength of sharply bounded induction. *Mathematical Logic Quarterly*, 52(6):613–624, 2006. doi:10.1002/malq.200610019.

13 Emil Jeřábek. Approximate counting in bounded arithmetic. *Journal of Symbolic Logic*, 72(3):959–993, 2007. doi:10.2178/jsl/1191333850.

14 Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers*. Springer, 2012. doi:10.1007/978-3-642-24508-4.

15 Ravi Kannan. Circuit-size lower bounds and non-reducibility to sparse sets. *Information and Control*, 55(1):40–56, 1982. doi:10.1016/S0019-9958(82)90382-5.

16 Jan Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1995.

17 Jan Krajíček. Small circuits and dual weak PHP in the universal theory of p-time algorithms. *ACM Transactions on Computational Logic* (TOCL), 22(2):1–4, 2021. doi:10.1145/3446207.

18 Jan Krajíček and Igor C. Oliveira. Unprovability of circuit upper bounds in Cook's theory PV. *Logical Methods in Computer Science*, 13(1), 2017. doi:10.23638/LMCS-13(1:4)2017.

19 Jan Krajíček, Pavel Pudlák, and Gaisi Takeuti. Bounded arithmetic and the polynomial hierarchy. *Annals of Pure and Applied Logic*, 52(1-2):143–153, 1991. doi:10.1016/0168-0072(91)90043-L.

20 Jan Krajíček. *Proof Complexity*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2019. doi:10.1017/9781108242066.

21 Alexis Maciel, Toniann Pitassi, and Alan R. Woods. A new proof of the weak pigeonhole principle. *Journal of Computer and System Sciences*, 64(4):843–872, 2002. doi:10.1006/jcss.2002.1830.

22 Moritz Müller and Ján Pich. Feasibly constructive proofs of succinct weak circuit lower bounds. *Annals of Pure and Applied Logic*, 171(2), 2020. doi:10.1016/j.apal.2019.102735.

23 Igor C. Oliveira. Meta-mathematics of computational complexity theory. *Preprint*, 2024.

24 Claude E. Shannon. The synthesis of two-terminal switching circuits. *The Bell System Technical Journal*, 28(1):59–98, 1949. doi:10.1002/j.1538-7305.1949.tb03624.x.

25 Bella A. Subbotovskaya. Realization of linear functions by formulas using $+$, $\cdot$, $-$. In *Soviet Math. Dokl*, 1961.

Quantum Advantage and Lower Bounds in Parallel Query Complexity

Joseph Carolan
University of Maryland, College Park, MD, USA

Amin Shiraz Gilani
University of Maryland, College Park, MD, USA

Mahathi Vempati
University of Maryland, College Park, MD, USA

Abstract

It is well known that quantum, randomized and deterministic (sequential) query complexities are polynomially related for total boolean functions. We find that significantly larger separations between the parallel generalizations of these measures are possible. In particular,

1. We employ the cheatsheet framework to obtain an unbounded parallel quantum query advantage over its randomized analogue for a total function, falsifying a conjecture of [Jeffery et al. 2017].
2. We strengthen 1 by constructing a total function which exhibits an unbounded parallel quantum query advantage despite having no sequential advantage, suggesting that genuine quantum advantage could occur entirely due to parallelism.
3. We construct a total function that exhibits a polynomial separation between 2-round quantum and randomized query complexities, contrasting a result of [Montanaro. 2010] that there is at most a constant separation for 1-round (nonadaptive) algorithms.
4. We develop a new technique for deriving parallel quantum lower bounds from sequential upper bounds. We employ this technique to give lower bounds for Boolean symmetric functions and read-once formulas, ruling out large parallel query advantages for them.

We also provide separations between randomized and deterministic parallel query complexities analogous to items 1-3.

2012 ACM Subject Classification Theory of computation → Quantum complexity theory

Keywords and phrases Computational complexity theory, quantum, lower bounds, parallel

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.31

Related Version *Full Version*: `https://arxiv.org/abs/2410.02665`

Funding *Joseph Carolan*: US Department of Energy grant no. DESC0020264.
Amin Shiraz Gilani: U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research in Quantum Computing and Quantum Testbed Pathfinder programs (award numbers DE-SC0020312 and DE-SC001904).
Mahathi Vempati: US Department of Energy grant no. DESC0020264 and the US Department of Energy Nuclear Energy University Programs award number DE-NE0009417.

Acknowledgements The authors thank Laxman Dhulipala for suggesting to investigate the relationship between quantum algorithms and parallelism, Luke Schaeffer and Chaitanya Karamchedu for helpful discussions, and Andrew Childs for valuable feedback on an earlier draft. We thank an anonymous reviewer for suggesting the idea for the function in Section 1.1.2, greatly simplifying our original construction. ASG additionally thanks Stacey Jeffery and Ronald de Wolf for helpful discussions.



© Joseph Carolan, Amin Shiraz Gilani, and Mahathi Vempati;
licensed under Creative Commons License CC-BY 4.0



1 Introduction

Quantum query complexity is a widely studied model for understanding the capabilities and limitations of quantum computers. Many of the important quantum algorithms and quantum-classical separation results are expressed in this model. For instance, the quantum period finding algorithm, a major ingredient in Shor's factoring algorithm [32], was first developed in the query model. There are many other examples of query problems for which quantum algorithms provably achieve exponential query advantage over their classical counterparts [33, 18].

However, these problems have partial domains, meaning they are not defined on most inputs. For functions whose domain includes all inputs (total functions), it is known that the quantum, randomized and deterministic query complexities are all polynomially related [10, 4]. Grover's seminal algorithm [24] for unstructured search achieves a quadratic speed-up over classical algorithms, and Ambainis et al. [6] and Aaronson et al. [3] construct functions that separate quantum and deterministic query complexities by a 4th power, and quantum and randomized query complexities by a 3rd power respectively.[1] These separations cannot be significantly improved in the sequential query model [10, 4].

A natural question is whether these algorithmic limitations are robust against generalizations of sequential query complexity. In this work, we consider a parallel model where an algorithm is allowed to make many queries at each step, as introduced in [27]. In particular, the p-parallel quantum query complexity, denoted by $\mathsf{Q}^{p\|}$, of a function is the minimum number of steps needed to compute it with bounded error, if at each step the algorithm is allowed to make p non-adaptive quantum queries.[2] We will also refer to this quantity by *p-query depth*, *p-query rounds* and *p-query layers* interchangeably. Notice that, for any function f and any parallelism p, $\mathsf{Q}(f)/p \leq \mathsf{Q}^{p\|}(f) \leq \mathsf{Q}(f)$ since any q-round p-parallel algorithm can be simulated by a pq-query sequential algorithm, and any q-query sequential algorithm can be simulated by a q-round p-parallel algorithm.[3]

This model is motivated by the fact that fault tolerant quantum computers must be inherently parallel: a large scale quantum computer which cannot do many gates at once cannot correct local errors faster than they occur. Therefore, quantum computers will necessarily be able to perform operations in parallel, meaning one should design algorithms to take maximum advantage of this capability. Further, near term quantum computations are limited to low depth due to short decoherence times. These factors make the parallel query model a more natural abstraction of real devices than a purely sequential model.

Prior work addressing parallel query complexity has indicated that quantum advantage tends to disappear as parallelism increases. Zalka [37] showed the parallel complexity for searching an unstructured list of size N is $\Theta\left(\sqrt{N/p}\right)$. This was generalized by Grover and Radhakrishnan [25], who proved that $\tilde{\Theta}\left(\sqrt{Nt/(p\cdot\min\{t,p\})}\right)$ p-parallel queries are necessary and sufficient to find t marked elements in a list of size N. Along with providing a systematic study of parallel quantum query complexity, Jeffery, Magniez and de Wolf [27] showed that the p-parallel quantum query complexity of k-SUM is $\tilde{\Theta}\left((N/p)^{k/k+1}\right)$.

---

[1] [3] originally showed a 2.5th power separation, which was boosted to a 3rd power via tight lower bounds on k-fold Forrelation shown in [8, 31].

[2] We define the deterministic parallel query complexity $\mathsf{D}^{p\|}$ and randomized bounded error parallel query complexity $\mathsf{R}^{p\|}$ similarly.

[3] Similar bounds hold for $\mathsf{R}^{p\|}$ and $\mathsf{D}^{p\|}$.

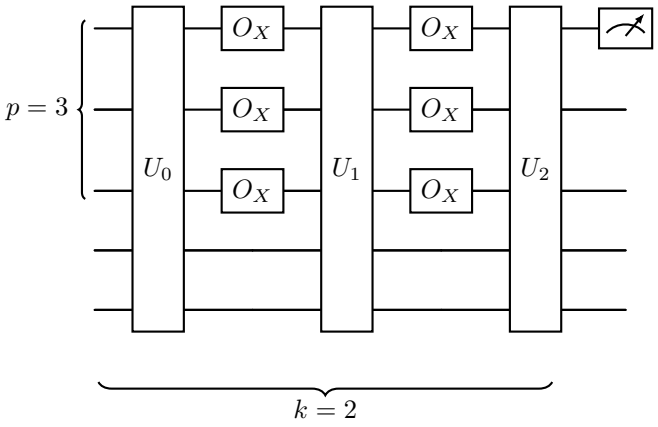


Figure 1 A parallel quantum algorithm with $k = 2$ adaptive steps and $p = 3$ queries per step; the last two wires are for workspace. In general, p denotes the number of non-adaptive queries per step, and k denotes the number of steps. The measure $Q^{p\|}$ denotes the minimal k needed to compute a function for a fixed p, while the measure $Q^{k\perp}$ denotes the minimal p needed to compute a function for a fixed k.

Furthermore, [27] introduce the following conjecture:

▶ **Conjecture 1** ([27]). *For any total function $f : \{0,1\}^N \to \{0,1\}$ and any p, there is at most a polynomial quantum advantage for p-parallel query algorithms. That is,*

$$\mathsf{D}^{p\|}(f) = \mathit{poly}(\mathsf{Q}^{p\|}(f)).$$

In fact, [27] proved this conjecture for all p polynomially smaller than the block sensitivity (bs), generalizing a result of Beals et al. [10].

The notion of p-parallel query complexity captures the minimum "depth" needed to compute some function for a given "width" (see Figure 1). An alternative complexity measure relevant to parallelism is the minimum "width" needed to compute some function for a fixed "depth", which we call the k-adaptive query complexity, and denote by $\mathsf{Q}^{k\perp}(f)$ (respectively $\mathsf{R}^{k\perp}(f)$, $\mathsf{D}^{k\perp}(f)$). Precisely, we define k-adaptive query complexity as the minimum p such that there is a p-parallel quantum (respectively randomized, deterministic) algorithm which makes k many p-parallel queries and computes f. This complexity measure is not widely studied for total functions. However, a result of Montanaro [28] fully characterizes the relevant 1-adaptive (or non-adaptive) query complexities, up to a factor of 2: for any total $f : \{0,1\}^N \to \{0,1\}$, $\mathsf{Q}^{1\perp}(f) \geq \mathsf{D}^{1\perp}(f))/2$. In the context of partial functions, a result by Girish et al. [22] characterized the maximal separation between $\mathsf{Q}^{k\perp}$ and $\mathsf{Q}^{k+1\perp}$.

In this paper, we present the high level overview of our work. See the full version on arxiv [14].

1.1 Main results

1.1.1 Unbounded parallel separations for total functions

Our first result involves adapting the cheatsheet framework of Aaronson et al. [3] to the parallel query setting, and using it to exhibit total function separations between parallel query complexity measures. In particular, we show that the canonical cheatsheet function of [3] exhibits unbounded separation between quantum and randomized parallel query complexities,

falsifying Conjecture 1. Further, we observe that this function demonstrates that 3-adaptive quantum algorithms can be more efficient in terms of total queries than even fully sequential randomized algorithms. More formally, we prove the following theorem.

▶ **Theorem 2.** *There exists a (total) Boolean function $h : \{0,1\}^N \to \{0,1\}$ such that for some $p = \tilde{O}(\mathsf{bs}(h))$ and some constants $\epsilon, \delta > 0$, we have*

(i) $\mathsf{R}^{p\|}(h) = \tilde{\Omega}(N^{\epsilon})$, **(ii)** $\mathsf{Q}^{p\|}(h) \leq 3$, **(iii)** $\mathsf{Q}^{3\perp}(h) = \tilde{O}\left(\mathsf{R}(h)^{1-\delta}\right)$.

It is worth noting that the large separation in parallel query complexity occurs when the parallelism almost exactly equals the block sensitivity $\mathsf{bs}(f)$, contrasting a result of Jeffery et al ([27], Theorem 12) which eliminates this possibility for $p = O(\mathsf{bs}(f)^{1-\epsilon})$ for any constant $\epsilon > 0$. Furthermore, the separation between $\mathsf{Q}^{3\perp}(f)$ and $\mathsf{R}(f)$ contrasts Montanaro's result ([28], Theorem 1) that for total functions, 1-adaptive quantum algorithms cannot be more than twice as efficient as the trivial 1-adaptive deterministic algorithm.

We also show a total function that achieves a similar separation between randomized and deterministic parallel query complexities. We do this by modifying the canonical cheatsheet function to instead embed a partial function with a large randomized-deterministic separation.

It is worth noting that this result is also sufficient to falsify Conjecture 1.

These results provide a recipe for achieving exponential parallel query advantages (from quantumness or randomness) for total functions by amplifying a polynomial sequential advantage for such functions. However, it is unclear whether an advantage can originate entirely from a quantum or randomized algorithm's ability to utilize parallelism more effectively than a randomized or deterministic algorithms. In other words, is there a function which has no sequential quantum query advantage, yet a large parallel quantum query advantage? We answer this question in the affirmative, even for total functions.

1.1.2 Unbounded genuine parallel separations

Our starting point is a partial function with the desired property. That is, we construct a partial function with (almost) equal randomized and quantum sequential query complexities, that does not admit any parallel advantage for randomized algorithms, yet admits near-maximal parallel advantage for quantum algorithms. In particular, we show the following theorem.

▶ **Theorem 3.** *There exists a (partial) Boolean function $f : \mathcal{D} \to \{0,1\}$ with $\mathcal{D} \subset \{0,1\}^N$ such that for some $p = O(\mathsf{Q}(f))$ and some constant $\epsilon > 0$, we have*

(i) $\mathsf{R}(f) = \tilde{\Theta}(N^{\epsilon})$, **(ii)** $\mathsf{Q}(f) = \tilde{\Omega}(\mathsf{R}(f))$, **(iii)** $\mathsf{R}^{p\|}(f) = \tilde{\Omega}(\mathsf{R}(f))$, **(iv)** $\mathsf{Q}^{p\|}(f) = 1$.

Note that the function f in the above theorem perfectly parallelizes quantumly but does not parallelize at all classically. We discuss the construction of f in more detail in the technical overview. We totalize f using the cheatsheet framework to give a total function with no sequential quantum advantage but unbounded parallel quantum advantage, giving rise to the following theorem.

▶ **Theorem 4.** *There exists a (total) Boolean function $h : \{0,1\}^N \to \{0,1\}$ such that for some p and some constants $\epsilon, \delta > 0$ (with $\delta < \epsilon$), we have*

(i) $\mathsf{R}(h) = \tilde{\Theta}(N^{\epsilon})$, **(ii)** $\mathsf{Q}(h) = \tilde{\Omega}(\mathsf{R}(h))$, **(iii)** $\mathsf{R}^{p\|}(h) = \tilde{\Omega}(N^{\delta})$, **(iv)** $\mathsf{Q}^{p\|}(h) \leq 3$.

We also obtain analogous separations between the randomized and deterministic query complexities using the same techniques.

1.1.3 Separations with two layers of adaptivity

We earlier noted (in Theorem 2) that for some total function h, $\mathsf{Q}^{3\perp}(h)$ can be polynomially smaller than $\mathsf{R}^{3\perp}(h)$ (and even $\mathsf{R}(h)$), however $\mathsf{Q}^{1\perp}(h) = \Theta(\mathsf{D}^{1\perp}(h))$ for all total functions h. A natural question to ask is whether it is possible for $\mathsf{Q}^{2\perp}(h)$ to be much smaller than $\mathsf{R}^{2\perp}(h)$ for some total h? We answer this question in the affirmative by presenting a general framework that transforms an (at least) polynomial separation between $\mathsf{Q}^{1\perp}(f)$ and $\mathsf{R}^{1\perp}(f)$ for some partial f to a polynomial separation between $\mathsf{Q}^{2\perp}(h)$ and $\mathsf{R}^{2\perp}(h)$ for some total h constructed from f. Since there exists a partial function f (such as Forrelation in [2]) with $\mathsf{Q}(f) = 1$ (so $\mathsf{Q}^{1\perp}(f) = 1$) and $\mathsf{R}(f) = \Omega(N^\epsilon)$ (so $\mathsf{R}^{1\perp}(f) = \Omega(N^\epsilon)$) for some $\epsilon > 0$, we get a total function h exhibiting a polynomial separation between $\mathsf{Q}^{2\perp}(h)$ and $\mathsf{R}^{2\perp}(h)$. Precisely, we show the following theorem:

▶ **Theorem 5.** *Let $f : \mathcal{D} \to \{0,1\}$ with $\mathcal{D} \subseteq \{0,1\}^N$ be a (partial) function that satisfies $\mathsf{Q}^{1\perp}(f) = \tilde{O}(N^\epsilon)$ and $\mathsf{R}^{1\perp}(f) = \tilde{\Omega}(N^\delta)$ for some constants $\epsilon < \delta$. Then there exists a total $h : \{0,1\}^M \to \{0,1\}$ (with $M = \Theta(N^3)$) and constants $\epsilon' < \delta'$ such that $\mathsf{Q}^{2\perp}(h) = \tilde{O}(M^{\epsilon'})$ and $\mathsf{R}^{2\perp}(h) = \tilde{\Omega}(M^{\delta'})$.*

We also obtain a similar separation between $\mathsf{R}^{2\perp}$ and $\mathsf{D}^{2\perp}$.

1.1.4 Quantum parallel lower bound framework and applications

Finally, we develop a new method for deriving parallel quantum query lower bounds from sequential quantum query upper bounds. It is simpler to reason about sequential algorithms than parallel algorithms, and upper bounds are often easier to give than lower bounds. Thus, our result allows reducing the hard problem of lower bounding parallel quantum query complexity to the potentially easier problem of upper bounding its sequential analogue. Our technique is based on the parallel spectral adversary method [27], and is especially well suited to problems with optimal adversary matrices that only distinguish input pairs that differ at a single index. We also show that, for any total function f, the combinatorial parallel adversary method [25, 13] fails to show a lower bound better than $\Omega\left(\sqrt{\left\lceil \frac{\mathsf{C}_0(f)}{p} \right\rceil \left\lceil \frac{\mathsf{C}_1(f)}{p} \right\rceil}\right)$ where C_b denotes the bth certificate complexity of f. Moreover, we show that our method surpasses this barrier and provides a better lower bound, for instance, to the And $\circ$ Or problem.

We let $\lambda(f)$ denote the spectral sensitivity of f, as defined in [4]. Let $\mathcal{F}^{(f)}_{p-\text{res}}$ denote the set of all functions obtained from restricting f to p input bits, and fixing the rest. We call this a $p-$restriction of f.

▶ **Theorem 6.** *For any total Boolean function $f : \{0,1\}^N \to \{0,1\}$, we have*

$$\mathsf{Q}^{p\|}(f) = \Omega\left(\lambda(f) \cdot \frac{1}{\max_{g \in \mathcal{F}^{(f)}_{p-res}} \lambda(g)}\right)$$

Recalling that $\lambda(f) = O(\mathsf{Q}(f))$, as shown by Aaronson et al. [4], we can write this as $\mathsf{Q}^{p\|}(f) = \Omega\left(\lambda(f) \cdot (\max_{g \in \mathcal{F}^{(f)}_{p-\text{res}}} \mathsf{Q}(g))^{-1}\right)$, where we note that the second term can be lower bounded by giving a (sequential) quantum algorithm for any $p-$restriction of f. We apply Theorem 6 to the following classes of well-studied functions.

1. *Read-once formulas*: While the combinatorial adversary method [5] is sufficient to lower bound the sequential quantum query complexity of the two-layer AND ∘ OR tree by $\Omega(\sqrt{N})$ [9], the best lower bound the parallel combinatorial adversary method can give for the Read-once formula problem $\mathsf{And}_{\sqrt{N}} \circ \mathsf{Or}_{\sqrt{N}}$ is $\Omega(\sqrt{N}/p)$. Using Theorem 6, we show a $\Omega\left(\sqrt{N/p}\right)$ lower bound for any read-once formula.
2. *Symmetric functions*: From Theorem 6, we get the tight lower bound of $\Omega\left(\sqrt{Nt/p \cdot \min\{t,p\}}\right)$ where t is the largest hamming weight less than $N/2$ such that the function value either differs on inputs of hamming weight t and $t+1$ or differs on inputs of hamming weight $N-t$ and $N-t-1$. This result recovers the combinatorial adversary lower bound implicit in [25] using an arguably simpler proof.

1.2 Related work

Prior work on parallel quantum algorithms in the query model has primarily demonstrated the need for quantum depth in solving certain problems.

Cryptography

There has been a recent line of work studying problems which can only be efficiently solved by quantum algorithms with high depth [15, 7, 16, 17, 21], constructing cryptographic proofs of quantum depth. In this vein, Chung et al [19] and Blocki et al [11] show parallel quantum lower bounds against producing the output of an iterated hash function to give a cryptographic proof of sequential work that is secure against quantum computers.

Partial functions

Burchard [13] gives a characterization of parallel quantum approximate counting, showing that, approximating to multiplicative error ϵ, the number K of marked elements in a list of size N requires $\Omega\left(\frac{\sqrt{N}}{\epsilon\sqrt{pK}}\right)$ p-parallel quantum queries. Girish et al [22] constructs a partial function which exhibits a large total query separation between r-adaptive and $r-1$-adaptive quantum algorithms for any constant r.

Circuit complexity

Another setting one could study the intersection of quantum and parallelism is in the circuit model. For instance, Cleve and Watrous [20] show how to implement the quantum fourier transform on n qubits in $O(\log n)$ depth, allowing for a highly parallel implementation of Shor's factoring algorithm. There are known examples of relational problems which can be computed in constant quantum depth, yet require at least logarithmic depth for classical circuits [12, 35, 36]. More generally, quantum circuit classes such as QAC_n and QNC_n have been extensively studied [23, 26, 34].

2 Technical overview

In this section, we outline the techniques used in showing our main results. For our results involving separations between query complexity measures, we will primarily describe the techniques involved in separating quantum from randomized query complexity measures unless the techniques for separating randomized and deterministic query complexity measures are significantly different.

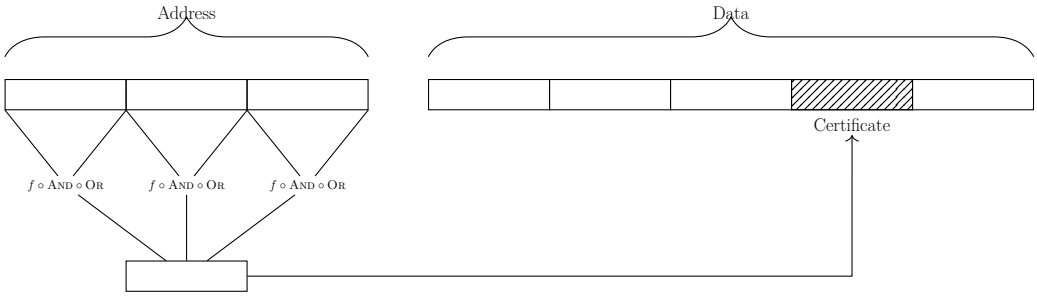


Figure 2 The canonical cheat sheet function f_{ccs} which lifts a partial function f to a total function, while retaining some of the speedup of f.

2.1 Unbounded parallel separations for total functions

The cheatsheet framework [3] lifts a partial function f, potentially with an exponential quantum query advantage to a total function that retains some of the advantage (although now polynomial). The natural way to do this is to define a total function such that when the input is in the domain of f, answer as f does, and when it is not answer 0. The cheat sheet framework does this "domain check" in a clever way so that the quantum advantage is not lost in the process. As shown in Figure 2, the input for the total function f_{ccs} is divided into two components, the Address section and the Data section. The partial function f is composed with a total function that has low certificate complexity, which in this case is $\textsc{And} \circ \textsc{Or}$. This composition $f \circ \textsc{And} \circ \textsc{Or}$ produces a single bit. This is repeated such that enough bits to address a block in the data component are obtained. If this block contains the certificate (or "cheatsheet") for the $\textsc{And} \circ \textsc{Or}$'s, and these certify that the input of f is in its domain, then the output of f_{ccs} is 1, else, the output is 0. As the data component is large, an algorithm is forced to solve the partial function f to obtain the address of the block, and can use the certificate present there to check whether the input of f is in the domain (and thus does not need to read the full Address to perform the domain check).

Thus, computing a cheatsheet function involves

1. Solving $f \circ \textsc{And} \circ \textsc{Or}$ to get the certificate location,
2. Reading out the certificate,
3. Checking the validity of the input of f.

For concreteness, consider the canonical cheatsheet function defined by Aaronson et al [3] composing $f = \textsc{Forrelation}$ [1, 2] on N bits with $\textsc{And} \circ \textsc{Or}$ on N^2 bits. This function was originally constructed to exhibit a superquadratic separation between sequential quantum and randomized query complexities. Observe that with $p = N^2$ parallelism, solving any given instance of the internal $\textsc{And} \circ \textsc{Or}$ can be done in a single p-parallel query by reading out all the inputs. A quantum algorithm can utilize this to quickly solve the whole composition, as Forrelation is quantumly easy. However, any classical algorithm would seem to need $\mathsf{R}(\textsc{Forrelation})$ p-parallel queries to solve this composition since it requires at least $p \cdot \mathsf{R}(\textsc{Forrelation})$ sequential queries. Thus, with p parallelism, step (1) is quantum efficient but not classically efficient. We show that for the canonical choice of parameters, steps (2) and (3) are also possible with few p-parallel queries, resulting in the desired separation.

The same technique can be used to give an exponential total function separation between randomized and deterministic parallel query complexity.

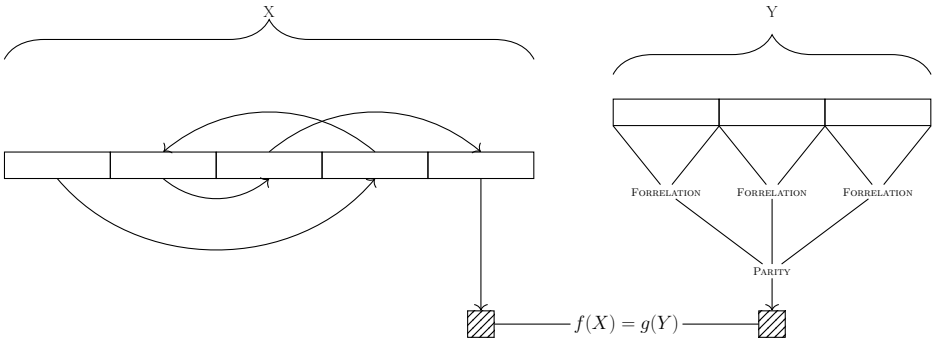


Figure 3 Partial function that admits unbounded genuine parallel separation.

Our result, combined with an aforementioned result of [27], which states that $\mathsf{D}^{p\|}(h) = \mathsf{poly}(\mathsf{Q}^{p\|}(h))$ for all total h and $p = O(\mathsf{bs}(h)^{1-\epsilon})$ with $\epsilon > 0$, provides a new way of upper bounding the block sensitivity of total Boolean functions. In particular, using our separation and a lower bound argument for the canonical cheatsheet function, we characterize its block sensitivity.

2.2 Unbounded genuine parallel separations

Is some polynomial sequential separation necessary for unbounded parallel quantum speedup, or can the advantage emerge purely from a quantum algorithm's ability to utilize parallelism (which we call a genuine parallel advantage)? We argue that the latter can indeed be the case, by first describing our construction for the partial function h that allows proving Theorem 3. As shown in Figure 3, the input to h here involves two components X and Y. The function f requires a sequence of adaptive queries to solve (in either model) whereas the function g achieves quantum-randomized separation and can be fully parallelized. We are promised that the output of both will be the same, so any algorithm can choose to solve either. We then argue that any algorithm will require solving at least one of f and g.

For concreteness, let $\epsilon > 0$ and suppose that $\mathsf{Q}(f) = \Theta(\mathsf{R}(f)) = \Theta(N^{\epsilon})$ and $\mathsf{R}^{p\|}(f) = \Omega(N^{\epsilon})$, and $\mathsf{Q}^{p\|}(g) = \Theta(N^{\epsilon})/p$ and $\mathsf{R}^{p\|}(g) = \Theta(N^{2\epsilon})/p$ for small enough p.[4] Then, a randomized sequential algorithm can choose to solve f, while a quantum sequential algorithm will not benefit from choosing to solve either f or g. Thus, $\mathsf{R}(h) = O(N^{\epsilon})$ and we would have $\mathsf{Q}(h) = \Omega(N^{\epsilon})$. Similarly, for $p = \mathsf{Q}(g)$, a p-parallel quantum algorithm can choose to solve g, while a p-parallel randomized algorithm will not benefit from choosing to solve either f or g. Therefore, $\mathsf{Q}^{p\|}(h) = 1$ and we would have $\mathsf{R}^{p\|} = \Omega(N^{\epsilon})$.

To show our desired randomized parallel and quantum lower bounds, we consider the general problem, which we call COR, where we are given inputs X and Y to arbitrary functions f and g respectively along with the promise that $f(X) = g(Y)$, our goal is to output $f(X) = g(Y)$. We use the hybrid argument to establish that any randomized parallel algorithm that solves $\textsc{cor}(f, g)$ will be able to either distinguish an input sampled from a hard 0-distribution for f from an input sampled from a hard 1-distribution for f, or will succeed at a similar distinguishing task for g. Therefore, we must have $\mathsf{R}^{p\|}(\textsc{cor}(f, g)) = \Omega(\min(\mathsf{R}^{p\|}(f), \mathsf{R}^{p\|}(g)))$. For the quantum lower bound, we show that that for any adversary

[4] For more concreteness, one may think of f as the pointer chasing function with chain length N^{ϵ} and g as the composition of parity on N^{ϵ} bits composed with FORRELATION on $N^{2\epsilon}$ bits as shown in Figure 3.

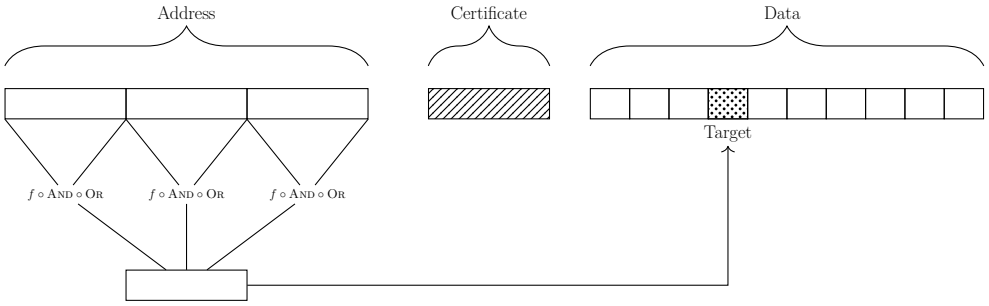


Figure 4 A framework that converts a partial function separation with one layer of adaptivity to a total function with two layers of adaptivity.

matrices $\Gamma^{(f)}$ and $\Gamma^{(g)}$ for f and g respectively, the matrix $\Gamma = \Gamma^{(f)} \otimes \Gamma^{(g)}$ is an adversary matrix for $\textsc{cor}(f,g)$ and $\max_i \|\Gamma_i\| = \max\left(\max_i \left\|\Gamma_i^{(f)}\right\|, \max_i \left\|\Gamma_i^{(g)}\right\|\right)$[5] It follows that $\mathsf{Adv}\left(\textsc{cor}(f,g)\right) = \Omega(\min(\mathsf{Adv}(f), \mathsf{Adv}(g)))$, which implies the desired lower bound.

Next, we describe a way to totalize the partial function h that we constructed in the previous section while maintaining some of its properties. We will use the cheatsheet framework. However, we will need a function with relatively small certificate complexity that does not admit any significant quantum speed-up so $\textsc{And} \circ \textsc{Or}$ will not work. Fortunately, [3] found a function, which they called Bkk, that satisfies $\mathsf{Q}(\textsc{Bkk}_N) = \tilde{\Theta}(\textsc{Bkk}_N) = \tilde{\Theta}(N)$ and $\mathsf{C}(\textsc{Bkk}_N) = \tilde{O}(\sqrt{N})$. We compose h on N bits with Bkk on N^2 bits, and then plug it into the cheatsheet framework. The desired upper bounds for $\mathsf{Q}^{p\|}$ and R in Theorem 4 are relatively straightforward and follows from the discussion in the previous sections. The quantum lower bound follows from quantum query complexity composition theorem and results in [3]. For the randomized parallel lower bound, we show a composition theorem where the inner function is Bkk.

The same lower bound techniques do not follow for the analogous separation between the randomized and deterministic query complexity measures. In particular, there is no general randomized composition theorem. Fortunately, for our constructions, the known randomized composition theorems suffice. For the deterministic parallel lower bound, we show a general composition theorem when the degree of the inner function is almost full, which might be of independent interest.

2.3 Separations with two layers of adaptivity

As we noted earlier (see Theorem 2), the cheat sheet framework can be employed to show a polynomial separation between $\mathsf{Q}^{3\perp}$ and $\mathsf{R}^{3\perp}$. We modify this framework to show a polynomial separation between $\mathsf{Q}^{2\perp}$ and $\mathsf{R}^{2\perp}$. As shown in Figure 4, our strategy is essentially to separate the cheatsheet into two parts: the first part contains the certificate, and the second just contains a long data string with one relevant index. Let f be a partial function with a polynomial separation between $\mathsf{Q}^{1\perp}(f)$ and $\mathsf{R}^{1\perp}(f)$. Informally, embedding f in our framework results in a function that requires (1) verifying the input to the f is in the domain (using the certificate) and if so, (2) outputting the bit of the data string present at the address obtained by solving the f.

[5] All the max are over relevant indices.

The quantum algorithm easily succeeds as follows. In the first round, it can read the certificate as well as solve f to obtain the address. In the second round, it verifies the certificate and queries the bit at the right address in the data string, hence is able to output the result. The randomized algorithm is unable to succeed in the same amount of parallelism because after the first round, it cannot compute f, so it would not find the right address by the first round, and can only make a query to the right address by the second round if it guesses the location correctly, which happens with very low probability. We also note that there is a caveat: the structure of a zero-certificate might be easy to distinguish from the structure of a one-certificate, allowing the algorithm to infer the inputs to the partial function from just the structure of the certificates. In that case, the randomized algorithm can use this to compute the partial function in the first round itself. To prevent this from happening, we use "bi-certificates" instead of certificates, where a "bi-certificate" corresponds to a set of indices that could certify both a zero and a one instance. This prevents the randomized algorithm from learning any information just by knowing the certificate.

Therefore, we present a framework to lift a partial function $\mathsf{Q}^{1\perp}$ vs $\mathsf{R}^{1\perp}$ separation to a total function $\mathsf{Q}^{2\perp}$ vs $\mathsf{R}^{2\perp}$ separation. The analogous result holds for randomized vs deterministic algorithms.

2.4 Quantum parallel lower bound framework and applications

There are few known techniques for lower bounding parallel quantum query complexity. Consider for instance lower bounding the quantum parallel query complexity for the balanced $\textsc{And} \circ \textsc{Or}$ function,

which is an example of a read-once formula. It is well known that this function has sequential quantum query complexity $\Theta(\sqrt{N})$ [9, 29]. Since both $\mathsf{Q}^{p\|}(\textsc{And})$ and $\mathsf{Q}^{p\|}(\textsc{Or})$ are $\Omega\left(\sqrt{N/p}\right)$, a natural guess for $\mathsf{Q}^{p\|}(\textsc{And} \circ \textsc{Or})$ is $\Omega\left(\sqrt{N/p}\right)$.

The bound $\mathsf{Q}(\textsc{And} \circ \textsc{Or}) = \Omega(\sqrt{N})$ was shown using the (sequential) combinatorial adversary method. However, as mentioned in Section 1.1.4, the corresponding parallel version fails to show a lower bound better than $\Omega\left(\sqrt{\left\lceil \frac{\mathsf{C}_0(f)}{p} \right\rceil \left\lceil \frac{\mathsf{C}_1(f)}{p} \right\rceil}\right)$ for any function f. This means that the best lower bound this method could help prove for $\textsc{And} \circ \textsc{Or}$ is $\Omega(\sqrt{N}/p)$, since $\mathsf{C}_0(\textsc{And} \circ \textsc{Or}) = \mathsf{C}_1(\textsc{And} \circ \textsc{Or}) = \sqrt{N}$.

We use the parallel spectral adversary method of [27], which is known to be optimal, to derive a method that is easier to apply and is sometimes stronger than the combinatorial adversary method (see Theorem 6). For the case of read-once formulas, we use the adversary sets of [9] to construct an adversary matrix Γ. It is easy to lower bound $\|\Gamma\|$ by a simple counting argument, but upper bounding $\|\Gamma_S\|$ for all $S \subseteq [N]$ with $|S| = p$ can be challenging. We show that if Γ satisfies the property that $\Gamma[x, y] = 0$ for all x, y that differs in more than 1 bit, then all the induced Γ_S can be rearranged to form block-diagonal matrices with blocks of size $2^p \times 2^p$. Moreover, each of these blocks are adversary matrices for some restricted function $g \in \mathcal{F}^{(f)}_{p-\text{res}}$ (Figure 5), where a restricted version of f is one where all but p input bits are fixed and known by the algorithm. Thus, we know that the spectral norm of these blocks is upper bounded by $\mathsf{Q}(g)$ (up to the normalizing factor of $\max_{i\in[2^p]} \|\Gamma_i\|$, and with a max taken over all $g \in \mathcal{F}^{(f)}_{p-\text{res}}$), which we can use to upper bound the spectral norm of any Γ_S. In our case, noting that g is a read-once formula of size p, we have $Q(g) = O(\sqrt{p})$ and we obtain the desired lower bound of $\Omega\left(\sqrt{N/p}\right)$. Hence, we are able to reduce the task of finding parallel lower bounds to the potentially easier task of finding sequential upper bounds.

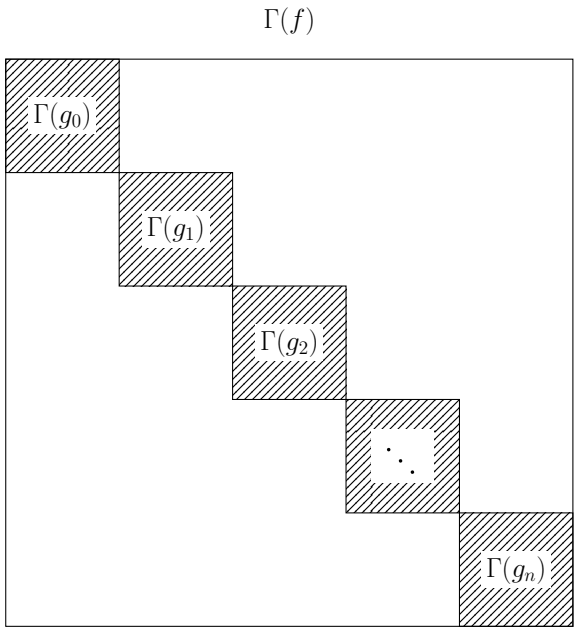


Figure 5 Nearest neighbor adversaries can be block-diagonalized, where the blocks are adversary matrices for restricted versions of the function f.

3 Conclusion

In this work, we have presented a few results on parallel quantum query complexity (See [14] for the full version). *A priori*, it seems unintuitive that an exponential advantage in number of rounds is possible for total functions, and also seems like lower bounding a simple function like AND ∘ OR in the parallel setting should be possible by standard techniques. Our results in Section 1.1.1 and Section 1.1.4 show that former is indeed true, and the latter is not the case and in fact requires "parallel-specific" thinking. Thus, one of our contributions is highlighting that there are still several fundamental query complexity questions unanswered in the parallel setting. We conclude by discussing a couple of these.

Limitations of parallel speedup

Can classical (randomized or deterministic) query algorithms simulate quantum query algorithms when allowed *both* a polynomial overhead in parallelism as well as number of rounds (as opposed to just a polynomial overhead in number of rounds as in Conjecture 1)? We conjecture that this is true:

▶ **Conjecture 7.** *For all total functions $f : \{0,1\}^M \to \{0,1\}$ and parallelism p, we have* $\mathsf{D}^{poly(p)\|}(f) = poly(\mathsf{Q}^{p\|}(f))$.

- When $p < \mathsf{bs}(f)^{1-\epsilon}$ for any $\epsilon > 0$, [27] prove that this is true.
- When $p \geq \mathsf{bs}(f) = \Omega(M^{\epsilon})$ for any $\epsilon > 0$, clearly this is true. This is the case in our result as well, where the canonical cheat sheet function was used to show Theorem 2 has $\mathsf{bs}(f) = \tilde{\Omega}(M^{1/6})$ and $\mathsf{D}^{p^{3/2}\|} = O(\mathsf{Q}^{p\|}(f)^3)$ for any p.
- Thus, the conjecture must be proven/falsified in the regime where bs and p are subpolynomial but superconstant. Moreover, what is the smallest power of p required to show this for all total f? Analogous questions remain open for $\mathsf{D}^{p\|}$ vs $\mathsf{R}^{p\|}$ as well.

Parallel composition theorem

It is known that $\mathsf{Q}(f \circ g) = \Theta(\mathsf{Q}(f) \cdot \mathsf{Q}(g))$ for boolean decision functions, a widely applicable and powerful result [30]. Is there an analogous theorem for p-parallel query complexity? In particular, do we have

$$\mathsf{Q}^{p\|}(f \circ g) = \Theta\left(\min_{q \in [p]} \mathsf{Q}^{q\|}(f) \cdot \mathsf{Q}^{\lceil p/q \rceil \|}(g)\right)$$

for boolean f, g? Such a result would suffice to reproduce our lower bound for $\textsc{And} \circ \textsc{Or}$, and likely be widely applicable for understanding parallel quantum query complexity. A straightforward attempt to show composition of the p-parallel adversary quantity using the reduction in [25] fails. This is because the partial function f' whose sequential query complexity characterizes the p-parallel complexity of f is no longer boolean.

Natural functions giving unbounded separation in rounds

Our results in Section 1.1.1 and Section 1.1.2 answer affirmatively whether it is possible for total functions to have unbounded separations in rounds, and whether this is still possible when no sequential separation exists. However, finding more natural functions to answer both these questions remains open and could help in understanding the interplay between quantum algorithms and parallel queries better.

References

1 Scott Aaronson. Bqp and the polynomial hierarchy. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, pages 141–150, 2010. doi:10.1145/1806689.1806711.

2 Scott Aaronson and Andris Ambainis. Forrelation: A problem that optimally separates quantum from classical computing. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, pages 307–316, 2015. doi:10.1145/2746539.2746547.

3 Scott Aaronson, Shalev Ben-David, and Robin Kothari. Separations in query complexity using cheat sheets. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, pages 863–876. Association for Computing Machinery, 2016. doi:10.1145/2897518.2897644.

4 Scott Aaronson, Shalev Ben-David, Robin Kothari, Shravas Rao, and Avishay Tal. Degree vs. approximate degree and quantum implications of huang's sensitivity theorem. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1330–1342, 2021. doi:10.1145/3406325.3451047.

5 Andris Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):750–767, 2002. doi:10.1006/jcss.2002.1826.

6 Andris Ambainis, Kaspars Balodis, Aleksandrs Belovs, Troy Lee, Miklos Santha, and Juris Smotrovs. Separations in query complexity based on pointer functions. *Journal of the ACM*, 64(5), 2017. doi:10.1145/3106234.

7 Atul Singh Arora, Andrea Coladangelo, Matthew Coudron, Alexandru Gheorghiu, Uttam Singh, and Hendrik Waldner. Quantum depth in the random oracle model. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pages 1111–1124, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3564246.3585153.

8 Nikhil Bansal and Makrand Sinha. K-forrelation optimally separates quantum and classical query complexity. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1303–1316. Association for Computing Machinery, 2021. doi:10.1145/3406325.3451040.

9 Howard Barnum and Michael Saks. A lower bound on the quantum query complexity of read-once functions. *Journal of Computer and System Sciences*, 69(2):244–258, 2004. doi:10.1016/j.jcss.2004.02.002.

10 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, 1998. doi:10.1145/502090.502097.

11 Jeremiah Blocki, Seunghoon Lee, and Samson Zhou. On the Security of Proofs of Sequential Work in a Post-Quantum World. In Stefano Tessaro, editor, *2nd Conference on Information-Theoretic Cryptography (ITC 2021)*, volume 199 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:27, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITC.2021.22.

12 Sergey Bravyi, David Gosset, and Robert Koenig. Quantum advantage with shallow circuits. *Science*, 362(6412):308–311, 2018. doi:10.1126/science.aar3106.

13 Paul Burchard. Lower bounds for parallel quantum counting. *ArXiv*, abs/1910.04555, 2019. arXiv:1910.04555.

14 Joseph Carolan, Amin Shiraz Gilani, and Mahathi Vempati. Quantum advantage and lower bounds for parallel query complexity, 2024. Full technical version of this paper, included in the QIP 2025 submission.

15 Nai-Hui Chia, Kai-Min Chung, and Ching-Yi Lai. On the need for large quantum depth. *J. ACM*, 70(1), January 2023. doi:10.1145/3570637.

16 Nai-Hui Chia and Shih-Han Hung. Classical verification of quantum depth, 2022. arXiv:2205.04656, doi:10.48550/arXiv.2205.04656.

17 Nai-Hui Chia and Shih-Han Hung. Non-interactive classical verification of quantum depth: A fine-grained characterization. Cryptology ePrint Archive, Paper 2023/1911, 2023. URL: https://eprint.iacr.org/2023/1911.

18 Andrew Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, pages 59–68, 2003. doi:10.1145/780542.780552.

19 Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao. On the compressed-oracle technique, and post-quantum security of proofs of sequential work. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, pages 598–629, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-77886-6_21.

20 R. Cleve and J. Watrous. Fast parallel circuits for the quantum fourier transform. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 526–536, 2000. doi:10.1109/SFCS.2000.892140.

21 Matthew Coudron and Sanketh Menda. Computations with greater quantum depth are strictly more powerful (relative to an oracle). In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pages 889–901, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3357713.3384269.

22 Uma Girish, Makrand Sinha, Avishay Tal, and Kewen Wu. The power of adaptivity in quantum query algorithms, 2023. doi:10.48550/arXiv.2311.16057.

23 Frederic Green, Steven T. Homer, Cristopher Moore, and Chris Pollett. Counting, fanout and the complexity of quantum acc. *Quantum Inf. Comput.*, 2:35–65, 2001. URL: https://api.semanticscholar.org/CorpusID:7821761, doi:10.26421/QIC2.1-3.

24 Lov Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, pages 212–219, 1996. doi:10.1145/237814.237866.

25 Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Symposium on the Theory of Computing*, 1996. URL: https://api.semanticscholar.org/CorpusID:207198067.

26 Peter Høyer and Robert Špalek. Quantum fan-out is powerful. *Theory of Computing*, 1(5):81–103, 2005. doi:10.4086/toc.2005.v001a005.

27 Stacey Jeffery, Frederic Magniez, and Ronald Wolf. Optimal parallel quantum query algorithms. *Algorithmica*, 79(2):509–529, 2017. doi:10.1007/s00453-016-0206-z.

28 Ashley Montanaro. Nonadaptive quantum query complexity. *Information Processing Letters*, 110(24):1110–1113, 2010. doi:10.1016/J.IPL.2010.09.009.

29 Ben W. Reichardt. Faster quantum algorithm for evaluating game trees. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 546–559, USA, 2011. Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611973082.43.

30 Ben W. Reichardt. Reflections for quantum query algorithms. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 560–569, USA, 2011. Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611973082.44.

31 Alexander A. Sherstov, Andrey A. Storozhenko, and Pei Wu. An optimal separation of randomized and quantum query complexity. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1289–1302, 2021. doi:10.1145/3406325.3451019.

32 Peter Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994. doi:10.1109/SFCS.1994.365700.

33 Daniel Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997. doi:10.1137/S0097539796298637.

34 Yasuhiro Takahashi and Seiichiro Tani. Collapse of the hierarchy of constant-depth exact quantum circuits. *computational complexity*, 25(4):849–881, 2016. doi:10.1007/s00037-016-0140-0.

35 Adam Bene Watts, Robin Kothari, Luke Schaeffer, and Avishay Tal. Exponential separation between shallow quantum circuits and unbounded fan-in shallow classical circuits. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 515–526, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3313276.3316404.

36 Adam Bene Watts and Natalie Parham. Unconditional quantum advantage for sampling with shallow circuits, 2024. doi:10.48550/arXiv.2301.00995.

37 Christof Zalka. Grover's quantum searching algorithm is optimal. *Phys. Rev. A*, 60:2746–2751, October 1999. doi:10.1103/PhysRevA.60.2746.

Succinct Fermion Data Structures

Joseph Carolan
University of Maryland, College Park, MD, USA

Luke Schaeffer
University of Waterloo, Canada

Abstract

Simulating fermionic systems on a quantum computer requires representing fermionic states using qubits. The complexity of many simulation algorithms depends on the complexity of implementing rotations generated by fermionic creation-annihilation operators, and the space depends on the number of qubits used. While standard fermion encodings like Jordan-Wigner are space optimal for arbitrary fermionic systems, physical symmetries like particle conservation can reduce the number of physical configurations, allowing improved space complexity. Such space saving is only feasible if the gate overhead is small, suggesting a (quantum) data structures problem, wherein one would like to minimize space used to represent a fermionic state, while still enabling efficient rotations.

We define a structure which naturally captures mappings from fermions to systems of qubits. We then instantiate it in two ways, giving rise to two new second-quantized fermion encodings of F fermions in M modes. An information theoretic minimum of $\mathcal{I} := \lceil \log_2 \binom{M}{F} \rceil$ qubits is required for such systems, a bound we nearly match over the entire parameter regime.

1. Our first construction uses $\mathcal{I} + o(\mathcal{I})$ qubits when $F = o(M)$, and allows rotations generated by creation-annihilation operators in $O(\mathcal{I})$ gates and $O(\log M \log\log M)$ depth.
2. Our second construction uses $\mathcal{I} + O(1)$ qubits when $F = \Theta(M)$, and allows rotations generated by creation-annihilation operators in $O(\mathcal{I}^3)$ gates.

In relation to comparable prior work, the first represents a polynomial improvement in both space and gate complexity (against Kirby et al. 2022), and the second represents an exponential improvement in gate complexity at the cost of only a constant number of additional qubits (against Harrison et al. or Shee et al. 2022), in the described parameter regimes.

2012 ACM Subject Classification Theory of computation → Quantum computation theory; Theory of computation → Data compression

Keywords and phrases quantum computing, data structures, fermion encodings

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.32

Related Version *Full Version*: `https://arxiv.org/abs/2410.04015`

Funding *Joseph Carolan*: supported by the US Department of Energy grant no. DESC0020264.

Acknowledgements The authors thank James Watson for helpful conversation about fermion encodings, and Andrew Childs for providing feedback on an early draft of this manuscript. LS thanks the Joint Center for Quantum Information and Computer Science (QuICS) where a large portion of this research occurred.

1 Introduction

The simulation of many interacting fermions is a promising application of quantum computers. Such systems quickly become difficult to simulate classically, with accurate simulation being computationally intractable for complex systems [28]. A quantum computer can perform highly accurate simulations with only polynomial resources for local systems, a wide and powerful class beyond what can be simulated classically.

Many quantum algorithms for fermionic simulation require a mapping from fermionic Fock states and creation/annihilation operators to qubit states and operators. Two of the most well known such mappings are the ones due to Jordan-Wigner [17] and Bravyi-Kitaev [7].



© Joseph Carolan and Luke Schaeffer;
licensed under Creative Commons License CC-BY 4.0



Table 1 Comparison to prior space efficient second-quantized encodings for F fermion, M mode systems, adapted from [19]. Gates refers to the complexity of implementing rotations generated by creation-annihilation operators. Here $\mathcal{I} := \left\lceil \log \binom{M}{F} \right\rceil$, which satisfies $\mathcal{I} = \Theta\left(F \log\left(\frac{M}{F}\right)\right)$.

Encoding	Qubits	Gates
Regime $F = o(M)$ and $F = \omega(1)$.		
Optimal Degree[19]	$\Omega(\mathcal{I}^2 \log^2 M)$	$\Omega(\mathcal{I}^2 \log^3 M)$
Qubit Tapering/Segment[6, 25, 24]	$M - o(M)$	$\Omega(F^2)$
Permutation Basis[14]	$\mathcal{I}$	$\Omega(M^2 2^{\mathcal{I}})$
Theorem 1	$\mathcal{I} + o(\mathcal{I})$	$O(\mathcal{I})$
Regime $F = \Theta(M)$		
Qubit Tapering/Segment[6, 25, 24]	$M - O(1)$	$\Omega(\mathcal{I}^2)$
Permutation Basis[14]	$\mathcal{I}$	$\Omega(\mathcal{I}^2 2^{\mathcal{I}})$
Theorem 2	$\mathcal{I} + O(1)$	$O(\mathcal{I}^3)$
General regime		
Theorem 1	$\mathcal{I} + O(F)$	$O(\mathcal{I})$
Theorem 2	$\mathcal{I} + O(\log(M/F))$	$O(MF\mathcal{I})$

These require M qubits to represent a fermion system with M modes, which is prohibitive in some cases. In particular, for particle preserving systems where the number of fermions F is much smaller than M, these encodings have a large amount of redundancy. In this case, the information theoretic lower bound of

$$\mathcal{I} := \left\lceil \log \binom{M}{F} \right\rceil \tag{1}$$

qubits is potentially much smaller than M.

Encodings which take advantage of this fact are relevant whenever the number of fermions is small compared to the number of modes, for instance in quantum chemistry simulations where the number of electrons (F) is fixed by a physical system yet the orbital basis size (M) should be as large as possible to correctly resolve the continuum behaviour. The discretization error in such systems scales like $1/M$ for reasonable bases [26], so large M is required for high accuracy – this is especially important when using basis sets that have not been classically optimized, such as the plane wave basis. This motivates developing simulation algorithms with as mild a dependence on M as possible in both space and gate usage, and a key component of many simulation algorithms is the qubit-to-fermion mapping used. Utilizing as few qubits as possible is important both for near term and fault tolerant devices, due to their limited size and the expense of qubits. However, space savings that incur an exponential gate overhead are not scalable. We therefore seek to minimize space complexity while still allowing relevant operations to be performed efficiently.

1.1 Prior work

There has been a line of work on second-quantized fermion encodings for saving space when the number of fermions is fixed. Bravyi et al. give a scheme for utilizing symmetries in particle preserving Hamiltonians to remove degrees of freedom, using parity symmetries to achieve $M - O(1)$ qubits generically and LDPC codes for few fermion systems to achieve $M - O(M/F)$ qubits [6], but with gate overhead $O(M^3)$ in the worst case[1]. Steudtner and Wehner give a

[1] Gate complexity here refers to the complexity of performing a single $\exp(-i\theta K)$, for some angle θ and number-preserving product of $O(1)$ many creation/annihilation operators K. For our discussions, we will not consider geometric locality.

scheme based on segmenting the space of fermions which again achieves $M - O(M/F)$ qubits, though allows more efficient $O(F^2)$ gate overhead [25, 24]. The same work also proposed a binary addressing code which shares some similarities with the encodings described here, but did not provide circuits for fermion operations nor bound the resource costs in the general case. The work of Babbush et al. [2] describes a way to store and manipulate few fermion systems using a sparse oracle, but do not provide explicit circuits for particle preserving rotations generated by creation-annihilation operators.

The work of Kirby et al. [19] describes a second quantized fermion encoding achieving $O(F^2 \log^4 M)$ space complexity and $O(F^2 \log^5 M)$ gate complexity, subject to a certain conjecture about Hermite interpolation. Additionally, the work of Harrison et al. [14] and Shee et al. [23] give methods for achieving exactly $\mathcal{I}$ qubit usage in second quantization, but at the cost of an exponential $M^{O(F)}$ gate complexity.

There are also first-quantized representations of fermion systems [26, 18, 1], where a list of occupied fermion modes is anti-symmetrized. This can save significant space when F is much smaller than M. However, such encodings allow for a different class of operations[2] to be performed efficiently as compared to second quantized encodings, making circuit complexity results within these two paradigms not directly comparable. Other works consider utilizing more than M qubits and restricting to an entangled subspace, allowing certain sets of k-local fermion operations to be extremely efficient (see e.g. [29, 30, 11, 7]). These encodings are most relevant for geometrically local systems, and in fact increase space beyond the standard Jordan-Wigner encoding – we therefore omit them from comparisons.

1.2 Contributions

We describe a quantum data structure which naturally captures the problem of encoding (second quantized) fermion systems in qubits. This structure represents bitstrings of length M and Hamming weight F, which naturally correspond to fermionic Fock states. We consider the (quantum circuit) gate/depth complexity of a sgn-rank ("sign rank") operator, which applies a phase conditioned on how many ones exist in some prefix of the bitstring, as well as a bit-flip operator which flips one of the bits. These operators allow us to straightforwardly express the fermionic operations which are relevant to quantum simulation, as we show in Section 3.

Within this framework, the Jordan-Wigner encoding can be seen as a data structure which encodes bitstring $b \in \{0,1\}^M$ by the quantum state $|b\rangle$, e.g. the trivial encoding. The sign rank operation is then simply a contiguous string of Z operators on some prefix, and the bit flip operator is a single X operator. Our encodings employ a different representation of bitstrings which is more efficient when F is small, requiring different quantum circuits.

1.2.1 Succinct structure

We first give a second quantized encoding for number preserving fermion systems which is nearly optimal in space usage, yet allows fermionic rotations with linear complexity and low depth. Our encoding uses sublinear redundancy, i.e. it is within a factor $1 + o(1)$ of optimal space usage (this is the meaning of "succinct" in the context of data structures) whenever $F = o(M)$. In particular, we prove the following theorem.

[2] In particular, first-quantized operations. See [26] for examples of usage of these encodings in quantum simulation.

▶ **Theorem 1** (Informal). *A system of F fermions in M modes can be represented by $\mathcal{I}+O(F)$ qubits such that particle preserving fermionic rotations*[3] *have circuits of complexity $O(\mathcal{I})$ and depth $O(\log M \log\log M)$.*

A summary of comparable prior work is shown in Table 1; notably, all previous second-quantized encodings either incur a polynomial overhead in space complexity[4], or the gate complexity of fermionic rotations is exponential in the number of qubits. We build up to this main result by presenting intermediary encodings achieving some, but not all of the aforementioned scalings. These encodings may be of independent interest due to their simplicity.

1.2.2 Implicit structure

We also give a construction which uses essentially the exact information-theoretic space minimum, except for an $O(1)$ number of ancilla (at constant filling). This construction achieves $\mathsf{poly}(M)$ gate complexity. The relevant regime for this encoding is constant filling, e.g. $F = M/3$ or more generally $F = \Theta(M)$. In this regime, the prior space complexity of $\mathcal{I} + O(F)$ may be a significant overhead, whereas when $F = o(M)$ the additive $O(F)$ term is asymptotically negligible. The information theoretic limit in the $F = \Theta(M)$ regime goes like $\mathcal{I} = \Theta(F)$, meaning there can be significant space to save if F is any constant fraction below $M/2$.

▶ **Theorem 2** (Informal). *A system of F fermions in M modes can be represented by $\mathcal{I} + O(\log(M/F))$ qubits such that particle preserving fermionic rotations have circuits of complexity $O(MF\mathcal{I})$.*

The most comparable prior work is that of Harrison et al. [14] and Shee et al. [23], which both achieve exactly $\mathcal{I}$ qubit complexity. However, these require an $\Omega\left(\binom{M}{F}\mathsf{poly}(M)\right)$ (e.g. exponential in M at constant filling) circuit complexity overhead for performing rotations generated by creation-annihilation operators, in the worst case. Our encoding improves exponentially on this gate complexity, achieving $\mathsf{poly}(M)$ in this regime. We do this at the cost of $O(1)$ ancilla qubits.

1.3 Technical overview

We begin in Section 3 by developing a data structure which naturally captures fermion to qubit mappings of number preserving systems. This definition is not fully general, but provides a framework for thinking about such mappings. We then instantiate this data structure in two ways.

1.3.1 First result: a succinct structure

To build up to the first main result, we begin by developing a fermion data structure in Section 4 based on a straightforward idea. Instead of storing the explicit string of ones and zeros, store a sorted list of pointers to the $\leq F$ positions containing a one. This approach builds

[3] By fermionic rotation, we mean the unitary $\exp(-i\theta K)$ for angle θ, where K is a number-preserving product of creation/annihilation operators, e.g. $K = a_j^\dagger a_k$ for some $j, k \in [M]$. We require that K is $O(1)$ local in the sense that it is the product of $O(1)$ creation/annihilation operators (but not geometrically local).

[4] i.e. have asymptotic space usage $\Omega(\mathcal{I}^c)$ for some $c > 1$ when $F \ll M$.

off ideas presented in prior fermion encodings [2, 25] as well as work compressing quantum states [8, 13], though we present these ideas in a way which will facilitate understanding our full encoding. Observe that, in contrast to first quantized representations [26] (which also store a list of pointers to occupied positions), we will not antisymmetrize the state. This enables us to efficiently implement sgn-rank and bit-flip queries (analogous to second-quantized rather than first-quantized operations), which we provide in Section 4.2. In Section 4.3 we describe a procedure using buffer registers that reduces the depth in this construction to logarithmic.

From this simple starting point, our next fermion data structure in Section 5 reduces the space requirements using combinatorial ideas, without affecting gate complexity. We accomplish this by splitting the most and least significant bits of each register, and using different representations for each. The most significant bits have a large amount of redundancy from being non-increasing, which we avoid by storing them using the stars and bars method. We give circuits for combining information between these two representations to efficiently implement the required operations.

We combine the two previous ideas with a succinct data structure that enables low-depth access to the most significant bits to show Theorem 1, with details given in the full version.

The key technical idea in this construction is a tree data structure used to store prefix sums related to the most significant bits. This prefix sum tree allows retrieving information about any specific entry in low depth, but requires less total space than a sorted list of numbers. This data structure therefore allows log depth fermionic rotations while being near-optimal in space usage, our first main result.

1.3.2 Second result: an implicit structure

Using a different approach, detailed in the full version, our second data structure achieves $\mathcal{I} + O(1)$ qubits and $\mathsf{poly}(M)$ gate complexity when $F = \Theta(M)$. We represent a bit string b (which itself represents a Fock state) by storing b's rank among feasible strings, where we determine rank by the number of bit-strings which are lexicographically before b or have a smaller Hamming weight than b. This encoding uses $\mathcal{I} + O(1)$ many qubits, but it is unclear how to interpret the encoded string to perform efficient operations. This obscurity is, essentially, the reason for the exponential gate overhead in prior works achieving this level of space efficiency.

We show how to perform both bit-flip and sign-rank efficiently on the first (unencoded) position, intuitively because the first and most-significant bit determines whether a given string has rank at most $\binom{M-1}{F-1}$ (first bit a 0) or rank above $\binom{M-1}{F-1}$ (first bit a 1) among strings of a given Hamming weight. This means that we can extract the first bit by comparing the encoded label against a fixed value, which can be done efficiently and with a small number of ancilla.

We then transform the ordering to one in which bit-flip and sign-rank can be performed efficiently on the second bit, and so on for each position. This is based on a construction of a certain set of orderings, which we denote $<_j$ for $j \in [0, ..., M]$, of the labels. These orderings have the property that, given just the rank of a configuration over order $<_j$, operations on the j-th encoded bit can be performed efficiently ($O(F\mathcal{I})$ gates) and with few ($O(1)$) ancillas on the label string. Further, we show how to transform a label under the order $<_j$ to a label under either $<_{j+1}$ or $<_{j-1}$ in the same complexity. By cycling over the M possible orderings, we can perform relevant operations with $O(MF\mathcal{I})$ gates and $O(1)$ ancillas.

2 Preliminaries

2.1 Fermions

A system of fermions with M modes can be defined by the algebra of creation ($a_i^\dagger$ for $i \in [M]$) and annihilation (a_i for $i \in [M]$) operators – this formalism is referred to as second quantization. These operators are determined by the anti-commutation relations

$$\{a_i^\dagger, a_j\} = \delta_{ij}, \qquad \{a_i^\dagger, a_j^\dagger\} = 0, \qquad \{a_i, a_j\} = 0. \tag{2}$$

Let $|00...0\rangle_f$ denote the vacuum state, the unique state which is not annihilated by any of the $a_i^\dagger$. We will primarily work with the Fock states, which are of the form

$$\begin{aligned} |\psi^{\mathbf{b}}\rangle_f :=& (a_1^\dagger)^{b_1}(a_2^\dagger)^{b_2}...(a_M^\dagger)^{b_M}\,|00...0\rangle_f \qquad \text{(for all } \mathbf{b} \in \{0,1\}^M\text{)} \\ =& |b_1, b_2, ..., b_M\rangle_f\,. \end{aligned} \tag{3}$$

We denote the span of these states $\mathcal{H}^{(fock)}$. Note that the M-mode Fock states are in natural correspondence with M-bit strings – this observation underlies the famous Jordan Wigner encoding [17], and will similarly underlie our data structures. It will be convenient for us to work in the Majorana basis determined by γ_j for $j \in [2m]$. These operators are defined as follows:

$$\gamma_{2j-1} = \frac{a_j^\dagger + a_j}{2}, \qquad \gamma_{2j} = \frac{i(a_j^\dagger - a_j)}{2}, \qquad \text{such that: } \{\gamma_j, \gamma_k\} = \delta_{jk}. \tag{4}$$

The action of Majorana operators on the Fock states can now be written as follows, where $\neg b$ is the negation of bit b,

$$\begin{aligned} \gamma_{2j-1}\,|b_1...b_j...b_M\rangle_f &= \left(\prod_{n=0}^{j-1}(-1)^{b_n}\right)|b_1...(\neg b_j)...b_M\rangle_f\,, \\ \gamma_{2j}\,|b_1...b_j...b_M\rangle_f &= i\cdot\left(\prod_{n=0}^{j}(-1)^{b_n}\right)|b_1...(\neg b_j)...b_M\rangle_f\,. \end{aligned} \tag{5}$$

In particular, these operators flip the occupation of a certain mode and apply a phase depending on the occupation of all preceding modes.

2.2 Fermion to Qubit Mappings

A fermion to qubit mapping can be defined by a linear mapping $\mathcal{E}_s$ from Fock states to qubit states, as well as a mapping $\mathcal{E}_o$ from Majorana operators to qubit operators. The qubit states/operators should be isomorphic to Fock states and Majorana operators, i.e. satisfy Equations 4, 5. Though some encodings do not fit into this paradigm (discussed in Section 1.1), this definition suffices for the encodings relevant to this paper. In particular, to define a mapping of an M mode system it suffices to construct $2M$ mutually anti-commuting qubit operators which each square to the identity.

We note that it is not always necessary to encode every possible Fock state, e.g. when simulating a system that does not explore every state. Particularly relevant for us will be particle preserving systems, which live in the subspace of Fock states with exactly F many fermions.

A well known example of a fermion-to-qubit mapping is the Jordan-Wigner encoding [17], which defines M-qubit operators as follows (where P_i denotes a Pauli P on the i-th qubit, acting trivially on the rest)

$$\mathcal{E}_{JW}(\gamma_{2i-1}) = \left(\prod_{j=1}^{i-1} Z_j\right) \otimes X_i, \qquad \mathcal{E}_{JW}(\gamma_{2i}) = \left(\prod_{j=1}^{i-1} Z_j\right) \otimes Y_i. \tag{6}$$

The mapped qubit operators in this encoding act on M qubits, which means M qubits are used to store fermionic states. Additionally, the circuit complexity of the mapped operators is $\Omega(M)$, as some mapped operators act non-trivially on all qubits.

2.3 Simulating Fermions

The main use case for fermion to qubit mappings is in the simulation of interacting fermions. In many such applications the dynamics are governed by a Hamiltonian H which is both k-local and term-wise number preserving (i.e. each term commutes with $\sum_i a_i^\dagger a_i$). If we wish to simulate such a system with M modes, the most general Hamiltonian is

$$H = \sum_{l\leq k; l \text{ even}; i_1 i_2 \ldots i_l \in [M]} h_{i_1 i_2 \ldots i_l} a_{i_1}^\dagger a_{i_2} \ldots a_{i_{l-1}}^\dagger a_{i_l} + h.c. \tag{7}$$

where the indices run over the M modes and $h.c.$ denotes the hermitian conjugate of the preceding term. This type of system is ubiquitous in quantum chemistry and physics. To simulate a fermionic system, one requires a mapping from fermion states/operators to qubit states/operators, as discussed in Section 2.2. Almost all second quantized algorithms for approximating evolution $\exp(-iHt)$ involve performing rotations generated by terms in H [9, 10, 4, 3, 20, 5, 21], so it is better for these rotations to require few gates and act on few qubits. Let V be a product of k Majorana operators, where k is even. Our encodings give efficient circuits for rotations of the form $\exp(-i\theta V)$ (k a multiple of 4) or $\exp(-\theta V)$ (k not a multiple of 4)[5]. This is sufficient to implement any rotation generated by the product of $O(1)$ creation/annihilation operators and its hermitian conjugate. For an illustrative example, consider a hopping term $a_j^\dagger a_k + h.c.$ for $j \neq k$,

$$\begin{aligned}\exp\Big(-i\theta(a_j^\dagger a_k + a_k^\dagger a_j)\Big) &= \exp(2\theta(\gamma_{2j-1}\gamma_{2k} - \gamma_{2j}\gamma_{2k-1})) \\ &= \exp(2\theta\gamma_{2j-1}\gamma_{2k}) \exp(-2\theta\gamma_{2j}\gamma_{2k-1}),\end{aligned}$$

from which it is clear that it suffices to implement two Majorana rotations. Note that although each Majorana operator itself need not preserve particle number, it changes the occupation number by at most k (in this case at most two). Further, the whole operator does preserve particle number. Therefore, so long as our encoding has enough "space" to fit the intermediate states, we will remain in the proper subspace after implementing the full operator $\exp\Big(-i\theta(a_j^\dagger a_k + a_k^\dagger a_j)\Big)$. We refer the reader to the full version for a more general discussion of this procedure.

[5] Actually, in most places we give circuits for the operator V up to a global phase. In the full version, we discuss how to generically implement the aforementioned rotations in the same complexity. Further, whether k is a multiple of 4 dictates whether $\exp(-i\theta V)$ or $\exp(-\theta V)$ is unitary; we implement the unitary one.

3 Fermion Data Structures

In this section we introduce an alternative perspective of fermion encodings, in a way which makes the connection to data structures explicit. This perspective will facilitate the development of efficient fermion-to-qubit mappings later on.

3.1 Definitions

We first define two useful combinatorial operations, sgn-rank and bit-flip. These are analogous to rank and bit flip queries respectively, which are well-studied in succinct classical data structures for bit-vectors [15, 22][6]. These operators act on a bit string $\mathbf{b} = (b_1, ..., b_M) \in \{0,1\}^M$ and depend on an index $j \in [M]$, and are combinatorial operations with no direct relation to fermions. However, they both show up naturally when writing down the action of fermionic creation-annihilation operators on Fock states: one can notice the similarities between Definitions 3, 4 and the action of Majorana operators in Equation 5.

▶ **Definition 3.** *The operator* sgn-rank *("sign rank") of an index $j \in [M]$ and bit string $\mathbf{b} \in \{0,1\}^M$ is the operator which returns the parity of the first j bits of $\mathbf{b}$ (represented as $+1$ for even, -1 for odd). In particular,*

$$\mathsf{sgn\text{-}rank}(j, \mathbf{b}) := \prod_{n=0}^{j} (-1)^{b_n}.$$

▶ **Definition 4.** *The* bit-flip *operator of an index $j \in [M]$ and bit string $\mathbf{b} \in \{0,1\}^M$ flips the j-th bit of $\mathbf{b}$. In particular, we have the following, where $\neg$ is logical negation,*

$$\mathsf{bit\text{-}flip}(j, \mathbf{b}) := (b_1, ..., (\neg b_j), ..., b_M).$$

With these two operations in hand, we can now describe the relevant type of (quantum) data structure for encoding fermionic states. Intuitively, we would like to represent a bit string $\mathbf{b} \in \{0,1\}^M$ in such a way that both sgn-rank and bit-flip queries can be implemented efficiently. Furthermore, in the settings we will care about we will be promised that the Hamming weight of $\mathbf{b}$ (the number of 1's) will never exceed $F + k$ or subceed $F - k$, for some $F < M$ and constant $k = O(1)$ (see the full version for more discussion). We use the notation $\mathcal{H}^{(fock)}_{M,F,k}$ to denote the Fock space of an M-mode system of fermions, restricted to states with occupation between $F - k$ and $F + k$: we call this range the capacity. We use the notation $\mathcal{H}^{(qubit)}_n$ to denote the Hilbert space of n qubits.

▶ **Definition 5.** *A* fermion data structure *of size M and capacity F, k is a qubit representation of Fock states with M modes and occupation at most $F + k$ and at least $F - k$, i.e. a linear and invertible mapping $\mathcal{E}_s : \mathcal{H}^{(fock)}_{M,F,k} \to \mathcal{H}^{(qubit)}_n$ that maps Fock states to computational basis states. We denote $\mathcal{E}_s(|\mathbf{b}\rangle_f)$ as $|\mathbf{b}\rangle$. We further require two n-qubit quantum circuit families F_j and R_j, answering* bit-flip *and* sgn-rank *queries respectively. Formally, we require*

$$F_j |\mathbf{b}\rangle := |\mathsf{bit\text{-}flip}(j, \mathbf{b})\rangle \qquad (\textit{if } \mathsf{bit\text{-}flip}(j, \mathbf{b}) \textit{ has } F - k \leq \mathsf{HW} \leq F + k), \tag{8}$$

$$R_j |\mathbf{b}\rangle := \mathsf{sgn\text{-}rank}(j, \mathbf{b}) |\mathbf{b}\rangle . \tag{9}$$

[6] Note that the differing models of complexity (RAM programs/cell probe versus quantum circuits) prevent most of these classical results from being directly applicable to our problems.

The gate/depth complexity of the above structure is the max gate count/depth, respectively, of the F_j, R_j over all $j \in [M]$. The space complexity is n – note that ancillas used in F_j, R_j are counted in this complexity, as they are circuits acting on exactly n qubits.

We remark that we consider quantum circuit complexity as the relevant cost measure, which prevents us from using many standard data structures results. Classical results often use the cell probe model [12], which is based on a RAM machine that allows random access to data. This model may not capture complexity in a real world quantum computer [16]. This necessitates new data structures and new ideas, which will be the focus of this paper.

3.2 Properties of fermion data structures

We first observe a subtlety in Definition 5: the operation of bit-flip is only required to be "correct" so long as the capacity is not exceeded. When one is interested in simulating a k-local Hamiltonian on a system with exactly F Fermions, then it often suffices to instantiate a Fermion data structure with capacity F, k[7]. In particular, it is straightforward to show that such a data structure can be used to instantiate a Trotterization scheme, or any simulation algorithm which relies on applying k-local particle-preserving rotations generated by products of k creation/annihilation operators. Intuitively, so long as we can "fit" the intermediate states necessary while performing these rotations, we will be able to perform any arbitrary sequence of rotations.

▶ **Proposition 6.** *Let $|\psi\rangle_f$ be a fermionic Fock state of F fermions in M modes, and C_f be a sequence of k-local fermionic rotations which preserve particle number, followed by a measurement in the Fock basis. If there is a fermion data structure of size M and capacity $F-k, ..., F+k$, using space n and gate complexity m, then there is a qubit state $|\psi\rangle$ on n qubits and a quantum circuit C of size $O(|C_f| \cdot m)$ such that measuring $C|\psi\rangle$ in the computational basis samples from the same distribution as $C_f|\psi\rangle_f$ (up to permuting bit-string labels).*

Proof. The initial state $|\psi\rangle$ is simply the encoding $\mathcal{E}_s(|\psi\rangle_f)$. To construct the circuit C, we will replace every gate of C_f (which are of the form $g_f = \exp(i\theta V)$ for k-local, particle preserving V) with the corresponding encoded circuit for performing the rotation, as detailed in Section 2.3. Let us call this encoded circuit g. Letting $|\phi\rangle$ be some superposition of Fock states with F fermions, then with a data structure $\mathcal{E}_s$ of capacity $F-k, ..., F+k$ we have the guarantee that

$$g\mathcal{E}_s(|\phi\rangle) = \mathcal{E}_s(g_f|\phi\rangle),$$

as g_f corresponds to a sequence of sgn-rank and bit-flip operations satisfying the necessary promises. Noting that $g_f|\phi\rangle$ will also have F fermions by the hypothesis that g_f preserves particle number, we can similarly perform this transformation for every following gate. Finally, by the invertibility of $\mathcal{E}_s$ we will obtain the same final output distribution by measuring in the computational basis, up to decoding the bit-strings (i.e. some permutation on the labels). ◀

Given the above discussion, the curious reader may note that our definition of fermionic data structures encodes all states of Hamming weight between $F-k$ and$F+k$ for some constant k. This is redundant in that $F+k > F$, and also that there are more states having Hamming weight between $F-k$ and $F+k$ than there are of Hamming weight exactly equal

[7] Note that this does incur some space overhead, but it is only an additive $O(1)$ (if $F = \Theta(M)$) or $O(\log M)$ (if $F = O(M/\log M)$)

to $F + k$. However, both of these redundancies result in negligible additive space overheads (either $O(1)$ or $O(\log M)$ depending on the regime). The conclusions we draw about space efficiency includes these overheads.

We remark that our encoding can also apply to some post-Trotter methods, e.g. certain linear combination of unitaries algorithms. We also remark for clarity that the encodings leading up to Theorem 1 will allow encoding every Fock state of occupation at most $F + k$ (though those leading up to Theorem 2 will only encode occupations $F - k$ through $F + k$).

4 Sorted List Fermion Data Structure

In this section, we describe a fermion data structure (Definition 5) based on storing a sorted list of pointers, rather than a full sparse string. While this encoding has some similarities to first-quantized encodings [26] and sparse oracle encodings [2], we emphasize that we present a generic second-quantized encoding capable of performing any $O(1)$-local particle preserving rotation generated by creation/annihilation operators in the described complexity. Furthermore, this encoding will lay the groundwork for our later, more efficient encodings.

▶ **Theorem 7.** *There is a fermion data structure for strings of M qubits having Hamming weight at most F that uses $O(F \log M)$ qubits, with gate complexity $O(F \log M)$.*

We will constructively establish this theorem throughout Section 4.1 and 4.2 – in particular Lemma 9 and Lemma 10 demonstrate the requisite circuit complexities. An immediate corollary is that a similar statement holds for fermion encodings.

▶ **Corollary 8.** *There exists a fermion encoding of an M mode system having at most F fermions which uses $O(F \log M)$ qubits such that any k-local, particle preserving rotation can be implemented with $O(F \log M)$ gates.*

4.1 Algebraic Definition

We formally define the state encoding function $\mathcal{E}_s^{(1)}$ and give the algebraic properties it satisfies. Let $e_i \in \{0,1\}^M$ be a binary string that is all 0's, except for a single 1 at position i (using 1-based indexing). Consider strings of the form

$$x = e_{i_1} \oplus e_{i_2} \oplus ... \oplus e_{i_f}$$
$$i_1 < i_2 < ... < i_f$$

with $f \leq F$. Define $\mathcal{E}^{(1)} : \mathcal{D} \to \{0,1\}^{F\lceil \log(M+1) \rceil}$, where $\mathcal{D} \subset \{0,1\}^M$ is the set of strings of Hamming weight at most F, as

$$\mathcal{E}^{(1)}(x) = |i_1\rangle |i_2\rangle \cdots |i_f\rangle |\infty\rangle \cdots |\infty\rangle \tag{10}$$

where ∞ is a placeholder for "no fermion", and is the larger of any comparison with a non-∞. Concretely, we adopt the convention that ∞ is represented by a register of all 1's. We can interpret the output as a sorted array of f elements, each element (also referred to as register) of size $\lceil \log(M+1) \rceil$, and padded out to F entries by ∞'s. We will now describe the action of sign-rank and bit-flip queries on this list.

1. A $\mathsf{sgn\text{-}rank}(j, \mathbf{b})$ query should apply a -1 phase if there are an odd number of registers having values less than or equal to j. Otherwise, it should act like the identity.
2. A $\mathsf{bit\text{-}flip}(j, \mathbf{b})$ should insert a register $|j\rangle$ into the list if it is not present, and otherwise delete $|j\rangle$. This operation should maintain sorted order, and act as described so long as both input and output states have Hamming weight less or equal to F.

Observe that both of the aforementioned operations are reversible. This suffices to define a fermion data structure.

4.2 Efficient Circuits

We will utilize comparison circuits between unsigned integers as a black box. As described in Section 4.1, we will assume an upper limit F on the number of pointers we will ever need to store, and therefore only utilize this many registers.

▶ **Lemma 9.** *For the sorted list encoding described in Section 4.1, a* $\mathsf{sgn\text{-}rank}(p, \mathbf{b})$ *query has a circuit of* $O(F \log M)$ *gates and* $O(F \log M)$ *ancilla.*

Proof. Such a circuit should induce a -1 phase for every 1 present (in the original bit string) before position p. To achieve this, for each pointer register $|x\rangle$ we compute $x \leq p$, apply a Z to the outcome bit, then uncompute. Each comparison takes $O(\log M)$ gates to compare $O(\log M)$ size numbers, and there are $O(F)$ comparisons to make; the overall circuit complexity is therefore $O(F \log M)$. ◀

▶ **Lemma 10.** *For the sorted list encoding described in Section 4.1, a* $\mathsf{bit\text{-}flip}(p, \mathbf{b})$ *query has a circuit of* $O(F \log M)$ *gates and* $O(\log M)$ *ancilla.*

Proof. Such a circuit should add a pointer p to the list if there is none, otherwise it should remove the pointer p. At a high level, we will do this in three steps.

1. If p exists, move it to the last register
2. On the last register exchange p and ∞
3. If the last register is p, move it to the sorted position

To accomplish this reversibly, we first implement a reversible ordered-swap U_p. This operator swaps two registers if one is p and the other is $> p$. By chaining these together in ascending order, we will move p to the end if it exists. Exchanging classical states can be easily done, then chaining more U_p together in descending order will move p to sorted order if it was at the end. The full circuit requires $O(F)$ calls to subroutine U_p, and each takes $O(\log M)$ (from comparisons); exchanging two classical values on a register has negligible complexity. The overall complexity is therefore $O(F \log M)$. ◀

4.3 Lower Depth

A notable problem with the previously described encoding is the serial nature of the bit flip operator. In particular, the depth of the circuits as described is $O(F \log M)$, primarily due to sequentially "bubbling" a targeted register to the end of the list when implementing a $\mathsf{bit\text{-}flip}(p, \mathbf{b})$ operation. In this section, we describe a modification using $O(F \log M)$ ancillas that allows exponentially smaller depth, $O(\log F + \log \log M)$. Further, the buffer register idea developed here will prove useful when lowering depth in our succinct construction.

To achieve $O(\log F + \log \log M)$ depth, we will pad the encoded state with an $|\infty\rangle$ between registers and on either end. We will refer to these as buffer registers, in contrast with logical registers representing pointers. We will also pad the start with a $|-\infty\rangle$ and the end with $|\infty\rangle$ logical registers for convenience. Formally, consider strings of the form

$$x = e_{i^{(1)}} \oplus e_{i^{(2)}} \oplus ... \oplus e_{i^{(f)}}$$
$$i^{(1)} < i^{(2)} < ... < i^{(f)}$$

where $f < F$. Define $\mathcal{E}^{(2)} : \{0,1\}^M \to \{0,1\}^{O(F \log M)}$ as

$$\mathcal{E}^{(2)}(x) = |-\infty\rangle\,|\infty\rangle\,|i_l^{(1)}\rangle\,|\infty\rangle\,|i_l^{(2)}\rangle\,|\infty\rangle \cdots |\infty\rangle\,|i_l^{(F)}\rangle\,|\infty\rangle \cdots |\infty\rangle\,, \tag{11}$$

where every other register is a buffer (coloured blue, but this is purely conceptual; the same data is stored in each). This leaves $O(F \log M)$ qubits, as the buffer registers are just a constant factor overhead. The correct action of sgn-rank and bit-flip exactly mirrors that of Section 4 on the logical registers (excluding the initial padded $-\infty$), except now we must also leave the buffer registers invariant after the computation is finished.

▶ **Lemma 11.** *Under the list-with-buffers encoding described in Section 4.3, a* sgn-rank$(p, \mathbf{b})$ *query has a circuit of* $O(\log \log M)$ *depth,* $O(F \log M)$ *gates, and* $O(F \log M)$ *ancilla.*

Proof. This operation can be done in a similar way to Section 4.2. We again apply a conditional phase to each logical register (note we do not do this on any buffer register, nor the padded $|-\infty\rangle$ at the beginning). The gate count $O(F \log M)$ follows from the analysis in Lemma 9, and one can see that the depth is fully determined by the depth of a comparison. By simple facts about comparator circuits, these can be done in $O(\log \log M)$ depth using linearly many ancillae, so the overall depth is $O(\log \log M)$. ◀

▶ **Lemma 12.** *Under the list-with-buffers encoding described in Section 4.3, a* bit-flip$(p, \mathbf{b})$ *query has a circuit of* $O(\log F + \log \log M)$ *depth,* $O(F \log M)$ *gates, and* $O(F \log M)$ *ancilla.*

Proof. We are promised that whenever an insertion is made, the last element will be an ∞. Conversely, whenever a deletion is made, the element added to the end will be an ∞. This prevents us from needing to perform a full cycle, instead relying on the buffer registers to provide ∞'s wherever needed. At a high level, it suffices to do the following:

1. Compute a flag f_{del} depending on whether p appears in the logical registers, and fan it out to all registers.
2. If p is not present ($f_{del} = \top$), insert p at the buffer register between it's immediate predecessor and successor.
3. If p is present, shuffle registers $> p$ upwards one position, and move p to it's preceding buffer. Otherwise, shuffle all registers $> p$ downwards one position, and move p out of it's preceding buffer.
4. If p was present ($f_{del} = \bot$), remove p from the buffer between p's predecessor and successor.
5. Uncompute f_{del}.

Note that steps (2) and (4) do not need to explicitly be controlled by f_{del} due to the assumed structure of the data. In step (3) however, the direction in which to shuffle depends on whether p is present or not, so a light cone argument implies that the local operations must depend on whether p is present. It is worth pointing out that, except for the fan-in/fan-out of f_{del}, all operations are nearest-register in a one dimensional layout, which could aid implementation on a quantum computer with geometric constraints.

Using linearly many ancilla registers, a comparison between n bit integers can be done in depth $\log n$. Swaps between registers can be done in $O(1)$ depth, the largest comparison is between $O(\log M)$ bit integers, and a single bit can be fanned out to F positions (one for each register) in depth $O(\log F)$. The total depth is therefore $O(\log F + \log \log M)$ (or $O(\log \log M)$ with unbounded fan-in and fan-out), using $O(F \log M)$ ancillae. Each register of size $\log M$ is acted on by only a constant number of comparison/swaps, so the gate complexity is $O(F \log M)$. ◀

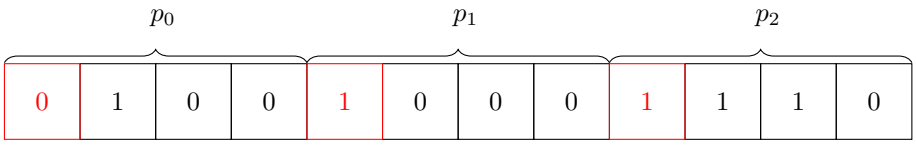


Figure 1 An illustration of the redundancy in a sorted list, for a list of 3 things out of a universe of 16 things. The red bits are non-decreasing, with possibilities: $000, 001, 011, 111$. Four possibilities means 2 bits suffice, rather than 3.

5 Achieving Succinctness

Recall that the information theoretic limit for encoding F fermions in M modes is $\mathcal{I} := \lceil \log \binom{M}{F} \rceil$ qubits. From Stirling's approximation we have

$$\mathcal{I} = F \log \frac{M}{F} + O(F),$$

so the encoding in Section 4.1 is redundant – for instance when $F = \sqrt{M}$, approximately half of the qubits are unnecessary. Here we present an encoding in which only a sublinear $O(F)$ many qubits are unnecessary (there are known constructions that achieve the *exact* information theoretic minimum $\mathcal{I}$ [14, 23], but the gate complexity becomes $M^{O(F)}$). For the example of $F = \sqrt{M}$, this new encoding has only a negligible $o(1)$ fraction of unnecessary qubits, i.e. it is roughly twice as space efficient as in Section 4. These techniques make crucial use of the fact that the list is not anti-symmetrized.

5.1 Succinct encoding

We formally state the main theorem of this section below.

▶ **Theorem 13.** *There is a fermion data structure for strings of M qubits having Hamming weight at most F that uses $\mathcal{I} + O(F)$ qubits, with gate complexity $O(F \log M)$.*

Such an encoding is constructed in the remainder of this section, with Lemma 15 and Lemma 16 demonstrating that it satisfies the requisite efficiencies. Similar to Corollary 8, this implies a fermion to qubit mapping with the same complexities.

▶ **Corollary 14.** *There exists a fermion encoding of an M mode system having at most F fermions which uses $\mathcal{I} + O(F)$ qubits such that any k-local, particle preserving rotation can be implemented with $O(F \log M)$ gates.*

To reduce the number of necessary qubits, the key insight is noting that the most significant $G := \lceil \log F \rceil$ qubits in each register of the encoding in Section 4.1 are not pulling their weight. They use $F \cdot G = \Omega(F \log F)$ qubits to encode F integers in the range 1 to F, but they are in *sorted order*, reducing the number of possible sequences – see Figure 1 for an illustration. We can instead store the same data (the most significant bits) in $O(F)$ space as a bit string where $2^G - 1 = O(F)$ zeroes delimit 2^G bins, and F ones correspond to F fermions. The i-th bin is the space from the i-th zero (or the start of the bit string) to the $i+1$-st zero (or the end of the bit string), and the number of ones in that range indicate elements with most significant bits i. This idea is commonly referred to as the stars and bars method.

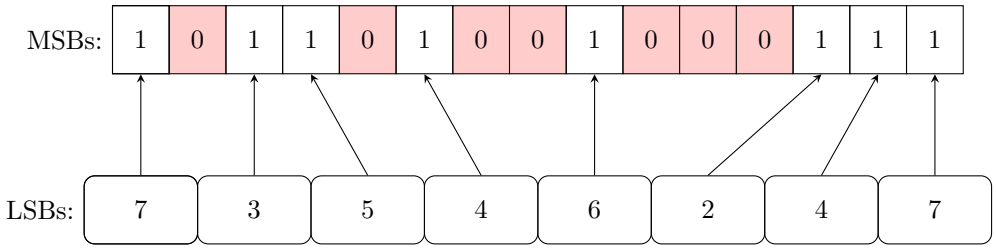


Figure 2 Illustrative example of the succinct representation, with parameters $F = 8$, $M = 63$ and fermions at $[7, 11, 13, 20, 38, 58, 60, \infty]$; recall the convention that the largest storable value (63) is identified with ∞. Red registers demarcate "bins" of different MSB values, out of 8 possible values. The LSBs then only require 3 qubits each, so the total storage is $15 + 24 = 39$ qubits. Using first quantization/sorted list encodings would use $8 \cdot \lceil \log 63 \rceil = 48$ qubits, and Jordan-Wigner or similar second quantized encodings would require 63 qubits.

The remaining least-significant bits can be stored in the usual way, where the order depends on the value determined by both least significant and most significant parts. So long as this order is maintained, the least-significant bits can be paired with the corresponding most-significant bits by the order in which they appear. This saves $F\lceil \log F \rceil$ qubits and adds $O(F)$ many, so the total qubit usage is

$$\begin{aligned}\text{Space}(M, F) =& F\lceil \log(M+1) \rceil - F\lceil \log F \rceil + O(F) \\ =& F \log \frac{M}{F} + O(F) \\ =& \mathcal{I} + O(F).\end{aligned}$$

To define the encoding formally, for strings of the form

$$\begin{aligned} x &= e_{i^{(1)}} \oplus e_{i^{(2)}} \oplus \ldots \oplus e_{i^{(F)}} \\ i^{(1)} &< i^{(2)} < \ldots < i^{(f)} \end{aligned}$$

for $f < F$, let i_m denote the G most significant bits of i, and i_l denote the remaining least significant bits. Let r_j be the number of pointers having most significant bits equal to j, i.e. the number of values of k such that $i_m^{(k)} = j$. Now define $\mathcal{E}^{(3)} : \{0,1\}^M \to \{0,1\}^{\mathcal{I}+O(F)}$ as

$$\mathcal{E}^{(3)}(x) = |i_l^{(1)}\rangle\, |i_l^{(2)}\rangle \cdots |i_l^{(f)}\rangle\, |\infty_l\rangle \cdots |\infty_l\rangle\, ||1^{r_0}01^{r_1}0 \ldots 01^{r_{2^G-1}} \tag{12}$$

Where $||$ is a conceptual divider between information about the most and least significant bits, and 1^n denotes n many 1's. The bits to the left of $||$ store the least significant bits of each pointer, and are referred to as l or LSBs, and i-th register in this section is l_i. The bits to the right of $||$ are referred to as m or MSBs and store the most significant bits of each pointer. Define $\mathcal{E}_s^{(3)} : \mathcal{H}_{2^M} \to \mathcal{H}_{2^{\mathcal{I}+O(F)}}$ as the unique linear extension of $\mathcal{E}^{(3)}$. A simple example of this encoding is shown in Figure 2.

For correctness, it suffices if encoded bit-flip and sgn-rank operators satisfy the rules described in Section 4.1, on the list implicitly represented by our succinct string. In particular, the encoded $\mathsf{bit\text{-}flip}(p, \mathbf{b})$ operator should unitarily insert/delete p from the sorted list defined by the information stored in the MSBs and LSBs, and update this data accordingly. The $\mathsf{sgn\text{-}rank}(p, \mathbf{b})$ operator should apply a minus phase if and only if there are an odd number of elements less or equal to p in the sorted list.

5.2 Efficient circuits

We now turn to constructing efficient circuits for the necessary operations on this succinct encoding. We note that retaining our space complexity advantage limits the number of ancillas that can be used to perform our operations to be $O(F)$ (and, naturally, we require these ancillas to be uncomputed by the end of each operation).

▶ **Lemma 15.** *Under the succinct list encoding $\mathcal{E}_s^{(3)}$ described in Section 5.1, a* $\mathsf{sgn\text{-}rank}(p, \mathbf{b})$ *query has a circuit of $O(F \log M)$ gates, and $O(F)$ ancilla.*

Proof. At a high level, this operation can be performed as follows:

1. Scan the most significant bits m to determine a range $[p_{start}, p_{end})$ of least significant bit registers whose most significant bits match p
2. Use this range to implement comparisons $\leq p$ on all registers
3. Phase the result of this comparison with a Z gate
4. Uncompute comparisons, then p_{start} and p_{end}

Scanning the register m takes $O(F)$ iterations costing $O(\log F)$ each. Each comparison afterwards costs $O(\log M/F)$ gates, and there are $O(F)$ many to make, so the total gate complexity is $O(F \log M)$. The only space used beyond comparisons are registers of size $O(\log F)$, satisfying the ancilla requirement. ◀

▶ **Lemma 16.** *Under the succinct list encoding $\mathcal{E}_s^{(3)}$ described in Section 5.1, a* $\mathsf{bit\text{-}flip}(p, \mathbf{b})$ *query has a circuit of $O(F \log M)$ gates, and $O(F)$ ancilla.*

Proof. Comparisons require information from the most significant bits, which is gathered first. Following this, analogous to the approach in Lemma 10 we will bubble the target element to the end if present, exchange with ∞, and then bubble back. At a high level:

1. Compute and store a pointer t to the target position in the list of F elements, defined as the first register with value $\geq p$ (based on both it's most and least significant bits).
2. Compute a flag f_{del}, which is on if and only if p is present in the list.
3. If f_{del}, shuffle l_t to the end of l.
4. Exchange $\infty_l \leftrightarrow p_l$ on the last register of l.
5. If not f_{del}, shuffle the end of l to t.
6. If f_{del}, shuffle the $(t+p)$-th bit of m to the end
7. If not f_{del}, shuffle the last bit of m to the $(t+p)$-th position.
8. Uncompute f_{del} and t.

Steps (1) and (2) can be done in $O(F \log M)$ gates by computing temporary p_{start}, p_{end} ranges of indices in l which have the same most significant bits as p, implementing comparisons using these, then uncomputing. Steps (3), (4), (6), (7) are simple swap ladders which can be performed in $O(F \log M)$ gates, and Step (5) has negligible complexity. Step (8) has the same cost as steps (1), (2), so the overall gate complexity is $O(F \log M)$. Again, the only additional space stored is $O(\log F)$ size pointers p_{start} and p_{end}, satisfying the ancilla requirement. ◀

6 Applications

In this section we give a more detailed comparison of our encoding to prior encodings, and outline scenarios where ours outperforms prior work. We will focus on the succinct construction, e.g. Theorem 1, as it is likely the more practical of the two constructions.

For concreteness we consider the parameter regime where the number of fermions is polynomially smaller than the number of modes, i.e. $F = M^c$ for some constant $0 < c < 1$, though we note that our encoding is efficient outside this regime as well. This regime is

well motivated for applications such as quantum chemistry, where the number of modes is preferred to be as large as possible to accurately model the continuum, but algorithm's polynomial scaling with M prevents M from being exponentially larger than F.

The primary application we find is in randomized simulation algorithms such as qDRIFT [9], where our encoding gives a polynomial space savings with only logarithmic depth overhead. We also show that our encoding allows for an optimal implementation of a second quantized SELECT subroutine, used in linear combination of unitary methods [10], no matter how M and F relate.

6.1 Space Efficient Randomized Simulation

Randomized product formulas like the qDRIFT algorithm [9] are strong candidates where our encoding will be useful, as the complexity of such algorithms depend heavily on the fermion encoding used. The qDRIFT algorithm requires $O((\lambda t)^2/\epsilon)$ fermionic rotation operators to evolve for time t within error ϵ, where λ is the sum of the magnitude of Hamiltonian coefficients when written in a second quantized basis. The scaling with λ is preferable to scaling with M when the magnitude of coefficients varies drastically, as is often the case for quantum chemistry Hamiltonians.

▶ **Theorem 17.** *For a particle preserving local fermionic Hamiltonian H on an M mode system, there is a second quantized algorithm for evolving an F fermion state to time t using space $\mathcal{I}+O(F)$, with gate complexity $O\left(\epsilon^{-1}(\lambda t)^2\mathcal{I}\right)$, and circuit depth $O\left(\epsilon^{-1}(\lambda t)^2 \log M \log\log M\right)$, where λ denotes the sum of magnitudes of coefficients in H and ϵ denotes the target diamond-norm error.*

This theorem follows straightforwardly from using the qDRIFT algorithm [9] with our Theorem 1 In the regime discussed above, the space complexity of this algorithm is

$$\mathcal{I} + O(F) = O(M^c \log M^{1-c}) = O(M^c \log M),$$

with gate complexity

$$O\left(\epsilon^{-1}(\lambda t)^2\mathcal{I}\right) = O(\epsilon^{-1}(\lambda t)^2 M^c \log M),$$

and circuit depth

$$O\left(\epsilon^{-1}(\lambda t)^2 \log M \log\log M\right).$$

We now compare these complexities against those achieved by other second-quantized encodings in our parameter regime.

6.1.1 Space Complexity

The most space-efficient prior second quantized encodings in this regime that are amenable to randomized simulation methods are the qubit efficient [23] and permutation basis [14], but these encodings would incur an exponential (M^F) gate and depth overhead. Barring these encodings, the next best are the segment code [25, 24], and the optimal degree code [19]. We use polynomially less space than the former ($O(M^c \log M)$ vs. $\Omega(M)$) and quadratically less than the latter ($O(F \log M)$ vs. $\Omega(F^2 \log^4 M)$).

6.1.2 Circuit Complexity

When $c < 1/2$, the prior best second quantized encoding in terms of space usage (barring codes with exponential gate overhead) is the optimal degree code [19]. Compared to this encoding, along with better space efficiencies we also improve on the $O(F^2 \log^5 M)$ gate complexity of fermion operations. Our scaling of $O(\mathcal{I})$ (with $\mathcal{I} < F \log M$) is always quadratically better in dependence on F, and at least four powers better in dependence on $\log M$. Further, while the depth is not analyzed in the optimal degree encoding a simple counting argument implies that it is at least $\Omega((\lambda t)^2 \log M/\epsilon)$, which is at most a negligible $O(\log \log M)$ better than our encoding. Compared to other space efficient second-quantized encodings, we give at least a quadratic improvement in circuit complexity while also removing many log factors.

Allowing space inefficient encodings, the most gate efficient prior encoding in our regime is the Bravyi-Kitaev encoding [7], which uses $O(\log M)$ gates to implement a fermionic rotation. This is the encoding used in the original resource estimates for the qDRIFT algorithm, making it a natural competitor. When the order of fermionic rotations is selected randomly, the Bravyi-Kitaev encoding of these operators cannot be significantly parallelized. The depth for full simulation will generically be $\Omega((\lambda t)^2/\epsilon)$ (and potentially larger if fermion operations take super-constant depth to implement), which is only a logarithmic factor lower than in our encoding. Furthermore, the Bravyi-Kitaev encoding requires polynomially more qubits ($O(M)$ versus $O(M^c \log M)$) than our encoding in this regime – though fewer gates ($O(\log M)$ versus $O(M^c \log M)$). This trade-off is therefore more beneficial the smaller c is.

6.2 Optimal SELECT Subroutine

In the context of post-Trotter methods, our encoding method can be relevant not only for saving space but also for saving gates. We illustrate this by giving a provably optimal implementation of the SELECT subroutine, a key component in the linear combination of unitaries algorithm for Hamiltonian simulation [10]. This algorithm involves expressing the Hamiltonian as a linear combination of unitaries

$$H = \alpha_1 U_1 + ... + \alpha_q U_q,$$

then block encoding the Hamiltonian using a PREPARE oracle that creates a superposition over every number in $j \in [q]$, weighted by amplitude α_j, in conjunction with as a so-called SELECT oracle. The SELECT oracle has a control wire indicating a number $j \in [q]$, and conditioned on j performs the unitary U_j on the physical state – this oracle is the only part whose implementation is dependent on the fermion encoding used. One way to split a k-local second quantized fermionic Hamiltonian into a linear combination of unitaries is to split every a_j into Majoranas $a_j = \gamma_{2j-1} + i\gamma_{2j}$. The γ_j operators are unitary, meaning the full Hamiltonian is now a linear combination of unitaries. Using our encoding, we can perform a SELECT oracle in the optimal complexity. Specifically, we show how to perform an k-local product of Majorana operators, given k index wires denoting which Majoranas appear in the product. This translates to a sequence of $O(k)$ many sgn-rank and bit-flip operations, so in particular it suffices to do a single $\textsf{sgn-rank}(j, \mathbf{b})$ or $\textsf{bit-flip}(j, \mathbf{b})$ operator controlled on an index wire $|j\rangle$.

▶ **Theorem 18.** *Using the encoding described in Theorem 1, we can perform a coherently controlled* $\textsf{sgn-rank}(|j\rangle, \mathbf{b})$ *or* $\textsf{bit-flip}(|j\rangle, \mathbf{b})$ *with* $O(\mathcal{I})$ *gates.*

Proof. Consider the M fermion Hamiltonian

$$H^{(m)} = \alpha_1 U_1^{(m)} + ... + \alpha_q U_q^{(m)}$$

where every $U_i^{(m)}$ is a product of Majorana operators. When the fermionic Hamiltonian H is k-local, then every $U_j^{(m)}$ appearing in $H^{(m)}$ has k-many Majorana operators. It follows that we can implement the action of $U_j^{(m)}$ within the encoding described in Theorem 1 with complexity $O(\mathcal{I})$. We can further coherently condition which $U_j^{(m)}$ is performed using a control register (see the full version for details), introducing at most a constant factor overhead. For this SELECT oracle we therefore obtain circuits of size $O(\mathcal{I})$. Note that in general there are many ways to split a Hamiltonian into a linear combination of unitaries, and we leave to future work whether SELECT has an efficient implementation under alternative choices. ◀

As argued above, this is sufficient to implement a SELECT oracle in the same complexity. We now turn to lower bounds.

▶ **Theorem 19.** *Using any fermion to qubit mapping, a coherently controlled, second-quantized, k-local particle preserving fermion operation (a SELECT oracle) requires $\Omega(\mathcal{I})$ gates.*

Proof. The lower bound for representing F fermions in M modes is $\mathcal{I}$ qubits, due to a straightforward counting argument. Suppose there were a generic implementation of the second quantized SELECT oracle of the aforementioned kind which used $\lfloor \frac{1}{2}(\mathcal{I}-1) \rfloor$ one- and two-qubit gates, and implemented any k-local fermion term for a $k > 1$. It follows that such an oracle would act on at most $\mathcal{I}-1$ qubits, so consider keeping only the $\mathcal{I}-1$ qubits it acts on. We are then left with a fermion encoding on $\mathcal{I}-1$ qubits, as by assumption the SELECT oracle is capable of performing arbitrary pairs of Majorana operators; a contradiction. Hence, any generic implementation of the SELECT oracle requires $\Omega(\mathcal{I})$ gates, matching our upper bound up to constant factors. ◀

To compare this result against existing second quantized encodings, note that in all prior encodings (summarized in Table 1) the complexity of a coherently controlled fermion operator is at least linear in the number of qubits, or exponential in the case of prior succinct encodings. For every gate-efficient encoding, the number of qubits is polynomially larger than $\mathcal{I}$ in our regime $F = M^c$, from which it follows that the gate complexity is at least polynomially larger than $\mathcal{I}$. Concretely, when $c < 1/2$ the best prior implementation of the select oracle is the optimal degree encoding, with gate complexity $\Omega(F^2 \log^5 M)$. Our implementation of complexity $O(\mathcal{I})$ where $\mathcal{I} < F \log M$ is quadratically better in F dependence, and at least four powers better in $\log M$ dependence. We note also that in the parameter regime $c > 1/2$ our encoding maintains at least a factor $\tilde{O}(M^{1-c})$ advantage in gate complexity over prior work, giving a polynomial advantage over the whole regime.

6.3 Comparison to First Quantized Encodings

First quantized algorithms maintain a list of pointers representing fermion positions similar to the encoding described in Section 4, but antisymmetrize the state over all possible permutations. The fermion phases then arise from the exchange statistics of the registers, allowing for a different class of efficient operations. Comparisons against first quantization are subtle, as algorithms within these two paradigms can have drastically different complexities for simulating similar systems. In general one would expect first quantized algorithms to

outperform second quantized algorithms when F is extremely small compared to M, but there is evidence that second quantized algorithms can remain advantageous when F is polynomially smaller than M [27]. This motivates the parameter regime we consider here.

6.3.1 Space Complexity

Using Stirling's approximation we can rewrite

$$\mathcal{I} = F \log \frac{M}{F} + O(F). \tag{13}$$

In our parameter regime of interest, we therefore have

$$\mathcal{I} = (1-c) \cdot F \log M + O(F) \tag{14}$$

Note that the $O(F)$ term is asymptotically less than the $(1-c) \cdot F \log M$ term, and as a corollary the asymptotic space usage of our encoding $(\mathcal{I} + O(F))$ is a factor $1 + o(1)$ from optimal, i.e. it is succinct. Standard first-quantized encodings [26] use $F\lceil \log M \rceil$, which is asymptotically a factor $\frac{1}{1-c}$ above $\mathcal{I}$, which is not succinct. Our encodings saves more space the closer c is to 1.

6.3.2 Gate Complexity

Both our representation and first-quantized representations support efficient (i.e. at most linear in the number of qubits) fermion operations of the first-quantized or second-quantized variety, respectively. Differences in gate complexity, therefore, stem primarily from differences in algorithms built in a first-quantized versus second-quantized picture. Many of the algorithms in these different paradigms are designed for different contexts, making a fair comparison difficult.

7 Conclusion and Open Problems

We presented two second-quantized fermion encodings which are almost exactly optimal in terms of space usage, yet allows gate efficient and low depth second-quantized fermion rotations. In the context of randomized simulation algorithms, the succinct encoding allows for a polynomial space savings with only a logarithmic depth overhead. Along the way, we presented a more simple succinct encoding which did not achieve low depth, as well as a low depth encoding that did not achieve succinctness, though both of these encodings may be of independent interest due to their simplicity. This work leaves open the following questions.

1. In first quantization, a list of positions is stored in an antisymmetrized wavefunction with $F\lceil \log M \rceil$ qubits in standard encodings, allowing for efficient first quantized fermion operations. Can such a representation be made succinct while still antisymmetrizing?
2. Is there an efficient fermionic Fourier transform circuit for our encodings?
3. Can the qubit count be improved beyond $\mathcal{I} + O(F)$ without exceeding gate complexity $O(\mathcal{I})$? We conjecture a lower bound on the qubits required to maintain this gate complexity is $\mathcal{I} + \Omega(F)$.
4. When compiled to a universal gate set like $\mathrm{CNOT}, \mathrm{H}, T$, the complexities of each gate is the same as the overall complexity in our encodings. In many error correcting codes certain gates are harder than others (for instance T gates are more expensive in a surface code), motivating analysis of each gate individually. Can the number of T gates be reduced in this construction?
5. Can the degree of the $\mathsf{poly}(M)$ circuit complexity scaling in our implicit encoding be improved?

References

1 Ryan Babbush, Dominic W. Berry, Jarrod R. McClean, and Hartmut Neven. Quantum simulation of chemistry with sublinear scaling in basis size. *npj Quantum Information*, 5(1):92, November 2019. doi:10.1038/s41534-019-0199-y.

2 Ryan Babbush, Dominic W Berry, Yuval R Sanders, Ian D Kivlichan, Artur Scherer, Annie Y Wei, Peter J Love, and Alán Aspuru-Guzik. Exponentially more precise quantum simulation of fermions in the configuration interaction representation. *Quantum Science and Technology*, 3(1):015006, December 2017. doi:10.1088/2058-9565/aa9463.

3 D. W. Berry, A. M. Childs, and R. Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 792–809, Los Alamitos, CA, USA, October 2015. IEEE Computer Society. doi:10.1109/FOCS.2015.54.

4 Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating hamiltonian dynamics with a truncated taylor series. *Phys. Rev. Lett.*, 114:090502, March 2015. doi:10.1103/PhysRevLett.114.090502.

5 Dominic W. Berry, Andrew M. Childs, Yuan Su, Xin Wang, and Nathan Wiebe. Time-dependent Hamiltonian simulation with L^1-norm scaling. *Quantum*, 4:254, April 2020. doi:10.22331/q-2020-04-20-254.

6 Sergey Bravyi, Jay M. Gambetta, Antonio Mezzacapo, and Kristan Temme. Tapering off qubits to simulate fermionic Hamiltonians, 2017. arXiv:1701.08213.

7 Sergey B. Bravyi and Alexei Yu Kitaev. Fermionic Quantum Computation. *Annals of Physics*, 298(1):210–226, May 2002. doi:10.1006/aphy.2002.6254.

8 Harry Buhrman, Bruno Loff, Subhasree Patro, and Florian Speelman. Memory compression with quantum random-access gates. In *Theory of Quantum Computation, Communication, and Cryptography*, 2022. URL: https://api.semanticscholar.org/CorpusID:247411405.

9 Earl Campbell. Random compiler for fast Hamiltonian simulation. *Phys. Rev. Lett.*, 123:070503, August 2019. doi:10.1103/PhysRevLett.123.070503.

10 Andrew M. Childs and Nathan Wiebe. Hamiltonian simulation using linear combinations of unitary operations. *Quantum Info. Comput.*, 12(11–12):901–924, November 2012. doi:10.26421/QIC12.11-12-1.

11 Charles Derby, Joel Klassen, Johannes Bausch, and Toby Cubitt. Compact fermion to qubit mappings. *Phys. Rev. B*, 104:035118, July 2021. doi:10.1103/PhysRevB.104.035118.

12 Anna Gál and Peter Bro Miltersen. The cell probe complexity of succinct data structures. *Theoretical Computer Science*, 379(3):405–417, 2007. Automata, Languages and Programming. doi:10.1016/j.tcs.2007.02.047.

13 Craig Gidney. Quantum dictionaries without QRAM, 2022. arXiv:2204.13835.

14 Brent Harrison, Dylan Nelson, Daniel Adamiak, and James Whitfield. Reducing the qubit requirement of Jordan-Wigner encodings of N-mode, K-fermion systems from N to $\lceil \log_2 \binom{N}{K} \rceil$, November 2022. arXiv:2211.04501.

15 Guy Joseph Jacobson. *Succinct static data structures*. PhD thesis, Carnegie Mellon University, USA, 1988. AAI8918056.

16 Samuel Jaques and Arthur G. Rattew. Qram: A survey and critique, 2023. arXiv:2305.10310.

17 P. Jordan and E. Wigner. On the Paulian prohibition of equivalence. *Z. Physik*, 47:631–651, 1928. doi:10.1007/BF01331938.

18 Ivan Kassal, Stephen P. Jordan, Peter J. Love, Masoud Mohseni, and Alán Aspuru-Guzik. Polynomial-time quantum algorithm for the simulation of chemical dynamics. *Proceedings of the National Academy of Sciences*, 105(48):18681–18686, 2008. doi:10.1073/pnas.0808245105.

19 William Kirby, Bryce Fuller, Charles Hadfield, and Antonio Mezzacapo. Second-quantized fermionic operators with polylogarithmic qubit and gate complexity. *PRX Quantum*, 3:020351, June 2022. doi:10.1103/PRXQuantum.3.020351.

20 Guang Hao Low and Isaac L. Chuang. Optimal hamiltonian simulation by quantum signal processing. *Phys. Rev. Lett.*, 118:010501, January 2017. doi:10.1103/PhysRevLett.118.010501.

21 Guang Hao Low and Isaac L. Chuang. Hamiltonian Simulation by Qubitization. *Quantum*, 3:163, July 2019. doi:10.22331/q-2019-07-12-163.

22 Gonzalo Navarro. *Compact Data Structures: A Practical Approach*. Cambridge University Press, 2016.

23 Yu Shee, Pei-Kai Tsai, Cheng-Lin Hong, Hao-Chung Cheng, and Hsi-Sheng Goan. Qubit-efficient encoding scheme for quantum simulations of electronic structure. *Phys. Rev. Res.*, 4:023154, May 2022. doi:10.1103/PhysRevResearch.4.023154.

24 Mark Steudtner. *Methods to simulate fermions on quantum computers with hardware limitations*. PhD thesis, Leiden University, 2019.

25 Mark Steudtner and Stephanie Wehner. Fermion-to-qubit mappings with varying resource requirements for quantum simulation. *New Journal of Physics*, 20(6):063010, June 2018. doi:10.1088/1367-2630/aac54f.

26 Yuan Su, Dominic W. Berry, Nathan Wiebe, Nicholas Rubin, and Ryan Babbush. Fault-tolerant quantum simulations of chemistry in first quantization. *PRX Quantum*, 2:040332, November 2021. doi:10.1103/PRXQuantum.2.040332.

27 Yuan Su, Hsin-Yuan Huang, and Earl T. Campbell. Nearly tight Trotterization of interacting electrons. *Quantum*, 5:495, July 2021. doi:10.22331/q-2021-07-05-495.

28 Matthias Troyer and Uwe-Jens Wiese. Computational complexity and fundamental limitations to fermionic quantum monte carlo simulations. *Phys. Rev. Lett.*, 94:170201, May 2005. doi:10.1103/PhysRevLett.94.170201.

29 F Verstraete and J I Cirac. Mapping local Hamiltonians of fermions to local Hamiltonians of spins. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09012, September 2005. doi:10.1088/1742-5468/2005/09/P09012.

30 James D. Whitfield, Vojtěch Havlíček, and Matthias Troyer. Local spin operators for fermion simulations. *Phys. Rev. A*, 94:030301, September 2016. doi:10.1103/PhysRevA.94.030301.

Bosonic Quantum Computational Complexity

Ulysse Chabaud
DIENS, ENS, PSL University, CNRS, INRIA, Paris, France

Michael Joseph
Department of Computer Science, Tufts University, Medford, MA, USA

Saeed Mehraban
Department of Computer Science, Tufts University, Medford, MA, USA

Arsalan Motamedi
Institute for Quantum Computing, Waterloo, Canada

Abstract

In recent years, quantum computing involving physical systems with continuous degrees of freedom, such as the bosonic quantum states of light, has attracted significant interest. However, a well-defined quantum complexity theory for these bosonic computations over infinite-dimensional Hilbert spaces is missing. In this work, we lay the foundations for such a research program. We introduce natural complexity classes and problems based on bosonic generalizations of **BQP**, the local Hamiltonian problem, and **QMA**. We uncover several relationships and subtle differences between standard Boolean classical and discrete-variable quantum complexity classes, and identify outstanding open problems. Our main contributions include the following:

1. **Bosonic computations.** We show that the power of Gaussian computations up to logspace reductions is equivalent to bounded-error quantum logspace (**BQL**, characterized by the problem of inverting well-conditioned matrices). More generally, we define classes of continuous-variable quantum polynomial time computations with a bounded probability of error (**CVBQP**) based on gates generated by polynomial bosonic Hamiltonians and particle-number measurements. Due to the infinite-dimensional Hilbert space, it is not a priori clear whether a decidable upper bound can be obtained for these classes. We identify complete problems for these classes, and we demonstrate a **BQP** lower bound and an **EXPSPACE** upper bound by proving bounds on the average energy throughout the computation. We further show that the problem of computing expectation values of polynomial bosonic observables at the output of bosonic quantum circuits using Gaussian and cubic phase gates is in **PSPACE**.
2. **Bosonic ground energy problems.** We prove that the problem of deciding whether the spectrum of a bosonic Hamiltonian is bounded from below is **co-NP**-hard. Furthermore, we show that the problem of finding the minimum energy of a bosonic Hamiltonian critically depends on the non-Gaussian stellar rank of the family of energy-constrained states one optimizes over: for zero stellar rank, i.e., optimizing over Gaussian states, it is **NP**-complete; for polynomially-bounded stellar rank, it is in **QMA**; for unbounded stellar rank, it is **RE**-hard, i.e., undecidable.

2012 ACM Subject Classification Theory of computation → Quantum information theory; Theory of computation → Quantum complexity theory

Keywords and phrases continuous-variable quantum computing, infinite-dimensional quantum systems, stellar rank, Hamiltonian complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.33

Related Version *Full Version*: `https://arxiv.org/abs/2410.04274` [8]

Funding S. M. and M. J. are grateful to the National Science Foundation (NSF CCF-2013062) for supporting this project. U.C. acknowledges support from the Plan France 2030 project NISQ2LSQ (ANR-22-PETQ-0006). U.C. and S.M. are grateful to have been supported by the Institute for Quantum Information and Matter, an NSF Physics Frontiers Center (NSF Grant PHY- 1733907), where preliminary ideas were discussed.



© Ulysse Chabaud, Michael Joseph, Saeed Mehraban, and Arsalan Motamedi;
licensed under Creative Commons License CC-BY 4.0



Acknowledgements U.C. acknowledges inspiring discussions with F. Arzani, R. I. Booth, F. Grosshans, D. Markham, O. Bournez, S. Lloyd, T. Vidick, W. Slofstra, L. Lami, and A. Winter. S.M. thanks A. Natarajan, P. Love, V. Podolskii, V.V. Albert, M. Hafezi, A. Gorshkov, D. Jacobs, and J. Jeang for inspiring discussions. A.M. acknowledges discussions with P. Sinha, Z. Mann, and R. Cleve. M. J. thanks V. Podolskii, D. Jacobs, W. Khangtragool and A. Saha for helpful exchanges. The authors are grateful to S. Aaronson for helpful discussions and to T. Vidick and F. Grosshans for comments on earlier versions of this manuscript.

1 Introduction

Many quantum mechanical systems, such as spin systems, are effectively described using discrete variables and are captured using qubits or qudits. The standard model of quantum computation is formulated based on such discrete degrees of freedom [37]. However, quantum variables such as the position, momentum, or amplitudes of electromagnetic fields are continuous. These degrees of freedom are described mathematically using infinite-dimensional Hilbert spaces. Continuous-variable systems are also present in many fundamental problems in theoretical physics, such as the solution to energy levels of a molecular system, quantum field theories, or exotic quantum states of matter such as Bose–Einstein condensates.

Continuous-variable quantum architectures have recently been used in practical quantum computing implementations, with several leading quantum error-correction schemes being fundamentally infinite-dimensional [20, 21]. Several important physical frameworks for quantum computing, such as quantum photonic processors, are based on continuous-variable degrees of freedom. These architectures have recently been demonstrated at scales sufficient to solve computational sampling problems that are believed to exceed the power of classical computations [46, 34].

What are the fundamental computational limitations and power of quantum degrees of freedom? This question is the subject of quantum complexity theory [44]. The standard quantum computing model and quantum complexity theory are formulated using discrete-variable quantum degrees of freedom over finite-dimensional Hilbert spaces. Several complexity classes have been defined to capture the computational power associated with these quantum degrees of freedom. One can define various natural models of quantum computations, such as adiabatic computation, quantum circuits, topological quantum computation, or measurement-based quantum computing, and all of these variants have been proven to have power equal to the computational complexity class **BQP**, which captures the power of polynomial-size quantum circuits in deciding logical statements with a small probability of error. Another important complexity class that captures the complexity of estimating the energy levels of a quantum physical system is **QMA**, which can be viewed as a quantum generalization of **NP** and, in particular, **MA** as the randomized generalization of **NP**. Due to a seminal result of Kitaev [28] (and many follow-up works, e.g., [29]), the problem of estimating the ground state energy of a physical system up to inverse-polynomial additive error (known as the local Hamiltonian problem) is complete for **QMA**. In fact, the local Hamiltonian problem can be viewed as the quantum generalization of the canonical constraint satisfaction problem over the Boolean hypercube, which is itself complete for **NP**. In recent years, the relationship between these complexity classes and standard complexity classes such as polynomial time **P**, polynomial space **PSPACE**, nondeterministic polynomial time **NP**, #**P** (corresponding to counting the number of solutions to constraint satisfaction problems), **PP** (the class of problems that are solvable on a probabilistic Turing machine with probability of error $< 1/2$) have been extensively studied. All of these complexity classes are contained in the class of

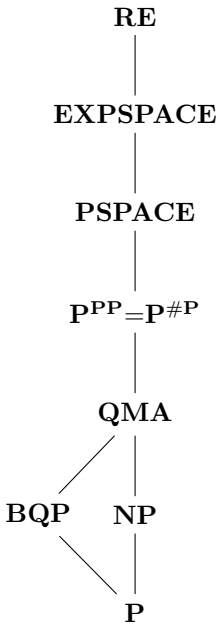


Figure 1 Known relations between complexity classes used in this work. If a line connects complexity class $\mathbf{C}_1$ to $\mathbf{C}_2$, with $\mathbf{C}_1$ being on the right of $\mathbf{C}_2$, it is implied that $\mathbf{C}_2 \subseteq \mathbf{C}_1$.

problems solvable in exponential space **EXPSPACE**, itself included in the set of recursively enumerable languages **RE**. The known relationships between these complexity classes are illustrated in Figure 1.

What are the basic relationships between the power of computations using bosonic continuous-variables over infinite-dimensional Hilbert spaces and these standard discrete-variable complexity classes over finite-dimensional ones? As it turns out, one can robustly encode discrete-variable computations into infinite-dimensional Hilbert spaces. For instance, Knill, Laflamme and Milburn showed that one can simulate arbitrary discrete-variable quantum computations using linear optics, particle-number measurements and feed-forward [32], i.e. adapting the computation based on the result of intermediate measurements, while Gottesman, Kitaev and Preskill gave a protocol to encode a qubit into the continuous degrees of freedom of a quantum harmonic oscillator in a fault-tolerant way [20]; the latter is among the leading proposals for reaching fault-tolerance in continuous-variable quantum architectures. As a consequence of these results, **BQP** can be robustly simulated using continuous quantum degrees of freedom. However, a formal complexity theory of bosonic computations is still missing.

In the continuous-variable setting, a model of quantum computation was first proposed by Lloyd and Braunstein [33]. In this model, the state of a quantum system evolves according to Hamiltonians, which are finite-degree polynomials in canonical continuous-variable (unbounded) operators such as the position operator X and the momentum operator P. These operators have fundamentally different mathematical properties from their discrete-variable counterparts. For instance, for any pair of discrete-variable operators acting on a finite-dimensional Hilbert space, A and B, $\mathrm{Tr}([A,B]) = 0$, it is well known that $[X,P] = iI$, which is not trace-class (i.e., does not have finite trace). The operators X and P have unbounded spectra with respective formal eigenbases $|x\rangle, x \in \mathbb{R}$ such that $X|x\rangle = x|x\rangle$, and $|p\rangle, p \in \mathbb{R}$ such that $P|p\rangle = p|p\rangle$, which are related to each other by a Fourier transform

$|x\rangle \propto \int e^{ipx} |p\rangle \, dp$, with their inner product giving Dirac delta functions $\langle x|x'\rangle = \delta(x - x')$, to be interpreted in the sense of distributions. Another striking example of the peculiarities of infinite dimensions comes from quantum states such as $(\sqrt{6}/\pi) \sum_{n\geq 1} \frac{1}{n} |n\rangle$, which are normalized but have infinite energy with respect to the harmonic oscillator Hamiltonian $X^2 + P^2$. Yet another famous example, known as Tsirelson's problem, is that the tensor product and the commuting operator formulations of quantum mechanics are equivalent in finite dimensions but not in infinite dimensions, because of complexity-theoretic reasons in particular [24].

Due to such nontrivial features, standard quantum complexity results in the discrete-variable setting, such as the Solovay–Kitaev theorem [31], do not have a clear counterpart in the continuous case. In particular, if one defines continuous-variable analogs of standard discrete-variable quantum computational complexity classes **BQP** and **QMA**, does it lead to relationships similar to the discrete-variable case?

Beyond its fundamental relevance, developing such a complexity-theoretic foundation for continuous quantum degrees of freedom would widen our horizon of knowledge in several ways. Firstly, it would help us to understand the computational power of continuous quantum degrees of freedom for real-world applications. Many physical architectures that are promising candidates for realizing a full-fledged universal discrete-variable quantum computer, such as photonic systems, superconducting qubits, and acoustic modes like phonons, are, in fact, inherently continuous. Developing a fundamentally continuous language for studying these systems could be essential to fully utilize the potential of such quantum computational devices, e.g., to devise fast compilation strategies [40] or performing Hamiltonian simulation with optimal constants [39].

Secondly, it touches on fundamental questions in theoretical physics. For instance, a standard approach to understanding continuous systems is to discretize them to a certain precision. We face an immediate question: can we reduce the study of these systems by approximating them with discrete variables, or are continuous-variable systems fundamentally distinct and require their own separate formulation? Theoretical physics provides some evidence for the latter. For instance, the renormalization group provides insight that the fundamental behavior of a physical system can vary significantly at different scales of precision. Or, in quantum field theory, one needs to impose energy cut-offs to avoid divergences in the computation of scattering amplitudes. Computational complexity studies the fundamental limits of finding reductions between two models, and we believe that tools from this framework would play a fundamental role in studying reductions between discrete and continuous degrees of freedom.

In this paper, we lay the groundwork for a computational complexity theory for quantum continuous variables. We formulate several Boolean quantum computational complexity classes based on continuous-variable bosonic generalizations of **BQP** and **QMA**. We study the relationship between these and their discrete-variable counterparts, highlight their fundamental similarities and differences, and identify outstanding open problems.

2 Main models and results

A bosonic quantum state can be described using a vector in the infinite-dimensional Hilbert space of square-summable complex sequences $\ell^2(\mathbb{C})$ which we refer to as a bosonic mode or a qumode. Let $\mathcal{H}_n \cong \ell^2(\mathbb{C}^{\otimes n})$ be the tensor-product space of n bosonic modes. We consider bosonic computations based on Hamiltonians H acting on $\mathcal{H}_n$ that are polynomials of degree d in the canonical operators X and P. Commonly known as position and momentum operators, they satisfy the canonical commutation relations

$$[X_a, P_b] = i\delta_{a,b} I, \quad [X_a, X_b] = 0, \quad [P_a, P_b] = 0. \tag{1}$$

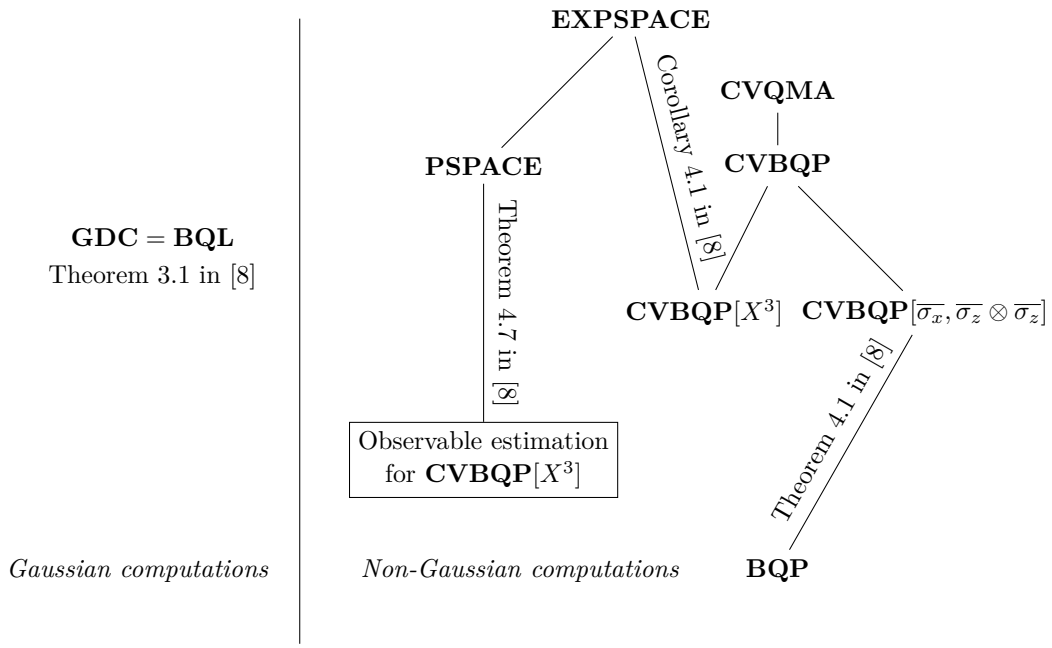


Figure 2 Relations between complexity classes and computational problems proven in this paper and described below, up to logspace reductions on the left and polynomial-time reductions on the right. If a line connects two complexity classes, it is implied that the one below in the diagram is included in the one above. If a line connects a problem to a complexity class, it is implied that the problem is in that class if it is below, and hard for that class if it is above. Square brackets indicate specific choices of non-Gaussian gates, where X^3 and $\overline{\sigma}_x$ are polynomial Hamiltonians of degree 3, while $\overline{\sigma}_z \otimes \overline{\sigma}_z$ is a polynomial Hamiltonian of degree 4. The other inclusions are given by the corresponding theorems or trivial from the definitions of the classes.

Polynomial Hamiltonians are ubiquitous in quantum physics and allow us to describe most existing models of bosonic quantum computations. The spectrum of such Hamiltonians is a subset of the reals, which can be discrete (e.g., for $X^2 + P^2$), continuous (e.g., for X) or both. In our presentation, we always distinguish between the cases of Gaussian Hamiltonians (polynomials of degree $d \leq 2$), which are typically less powerful, and non-Gaussian Hamiltonians (polynomials of degree $d > 2$).

We will often make use of the particle number operator (or simply, number operator) defined in terms of the position and momentum operators by,

$$N := \frac{X^2 + P^2 - I}{2} \tag{2}$$

where $N|n\rangle = n|n\rangle$ for the Fock basis states $|n\rangle$. The number operator plays a key role in our analysis as it quantifies the quanta of excitation, or 'particles,' in a bosonic mode. In terms of the energy of a bosonic state, we often refer to the average particle number, which corresponds to the expectation value $\langle N \rangle$.

2.1 Structure and summary of results

The structure of the full version of the paper [8] is as follows. After preliminary background in Section 2, we first consider Boolean complexity classes based on bounded-error continuous-variable quantum computations in Section 3 (Gaussian) and Section 4 (non-Gaussian). These contributions are summarised in Figure 2.

- We define a complexity class corresponding to the power of Gaussian dynamical computations (**GDC**) with logspace preprocessing. We prove that the power of this model is equivalent to the complexity class **BQL** corresponding to quantum logspace (See Theorem 3.1 in [8]). A complete problem for this class is inverting well-conditioned matrices [16, 42]. This can be viewed as a continuous-variable version of a result by Aaronson and Gottesman [2], which places the problem of simulating Clifford circuits in the complexity class $\oplus\mathbf{L}$ (a complete problem for this class is computing the determinant over the finite field $\mathbb{Z}_2$ [12]).
- We then define bounded-error quantum computations using gates generated by polynomials in the bosonic position and momentum operators (**CVBQP**). In this model, we sample the output of the quantum computation in the particle number basis (eigenstates of $X^2 + P^2$). We show that a specific instance of this class contains **BQP** (Theorem 4.1 in [8]) and that continuous-variable quantum computations using only Gaussian and cubic phase gates (generated by X^3) can be strongly simulated in **EXPSPACE** when no energy upper bound is assumed (Theorem 4.2 in [8]). We explain why proving a stronger upper bound of **PSPACE** or even **PP** as in the discrete-variable case (if true) would require nontrivial ideas. Finally, we give a polynomial-time parallel algorithm for computing output continuous-variable observable expectation values of such computations (Theorem 4.7 in [8]), which is in **PSPACE** by standard results [43]. We explain in Section 4.3.3 of [8] that the **EXPSPACE** and **PSPACE** results for particle-number measurements v.s. continuous-variable observable measurements are due to fundamental computational differences in Schrödinger v.s. Heisenberg dynamical evolution in continuous-variable systems. We conjecture that a strong simulation of the former model is strictly more difficult in terms of computational complexity.

Next, we consider ground state energy problems and Boolean non-deterministic quantum complexity classes in Section 5. These contributions are summarised in Table 1:

- We show that deciding the boundedness of the ground state energy is in **P** for Gaussian Hamiltonians (Theorem 5.1 in [8]) and **co-NP**-hard for constant-degree polynomial Hamiltonians (Theorem 5.2 in [8]). Furthermore, we give an efficient classical algorithm for verifying the boundedness of the ground state energy of a subclass of Hamiltonians of degree $d = 4$ via a reduction to a classical sum-of-squares method (Proposition 5.4 in [8]).
- We study the continuous-variable local Hamiltonian problem ($\mathsf{CVLH}^d_{\mathcal{S}}$) of estimating the lowest energy of a bosonic Hamiltonian of degree d in the position and momentum operators over a set of states $\mathcal{S}$. For Gaussian Hamiltonians, we show that this problem is in **P** whenever the set $\mathcal{S}$ contains the set of Gaussian states (Theorem 5.1 in [8]). For non-Gaussian Hamiltonians, we prove that the complexity of this problem critically depends on the stellar rank r, a measure of the non-Gaussian character of a continuous-variable quantum state [9, 10]. In order to simplify the presentation of the second result below, we rely on a mathematical conjecture (Conjecture 2 in [8], for which we provide numerical evidence), which allows us to parametrize the relevant family of states one optimizes over using simple constraints on the energy (average particle number):
  - For $r = 0$ (corresponding to optimization over Gaussian states) with at most $\mathsf{exp}(n) := e^{n^{O(1)}}$ energy (average particle number), we prove that the problem is **NP**-complete using a reduction from deciding when a matrix is not copositive (Theorem 5.3 in [8]).
  - When energy and stellar rank are both at most $\mathsf{poly}(n) := n^{O(1)}$, we prove that the problem is in **QMA** (Theorem 5.4 in [8]). The same proof technique also shows that for arbitrary r and at most $\mathsf{exp}(n)$ energy (average particle number), the problem is in **NTIME** $(n^{O(r)})$, where **NTIME** (t) is the class of problems that are solvable by a nondeterministic Turing machine that runs in time $O(t)$ on each branch.

- For $r = \infty$ with no energy bound, using an observation of [30], we encode the solution to Hilbert's tenth problem in the ground state of a Hamiltonian. As a consequence, the problem is **RE**-hard (undecidable) when we assume no bound on the stellar rank (Theorem 5.6 in [8]).

- Finally, we introduce a continuous-variable version of **QMA**, based on a **CVBQP** verifier (**CVQMA**). We give preliminary ideas for relating the complexity of the continuous-variable local Hamiltonian problem to this class, based on a continuous-variable analog of Kitaev's history state construction [29, 28].

Table 1 Computational complexity of the Hamiltonian boundedness problem and the continuous-variable local Hamiltonian problem for Gaussian and non-Gaussian polynomial bosonic Hamiltonians over n modes. $\mathcal{S}^{\mathsf{exp}}_{O(1)}$ denotes the set of states with constant stellar rank with at most exponential energy (average particle number) and $\mathcal{S}^{\mathsf{poly}}_{\mathsf{poly}}$ the set of states with polynomially bounded stellar rank and energy. Note that Theorem 5.4 and Theorem 5.5 of [8] rely on Conjecture 2, for which we provide numerical evidence. For all non-Gaussian results, assuming a degree $d = 4$ is sufficient except for the undecidability result where we show that $d = 8$ is sufficient (for $d = 4$, the problem is **QMA**-hard as a consequence of [11]).

Ground state problems	Gaussian Hamiltonians	Non-Gaussian Hamiltonians
HBound	$\in$ **P** (Theorem 5.1 in [8])	**co-NP**-hard (Theorem 5.2 in [8])
CVLH over $\mathcal{S}^{\mathsf{exp}}_{0}$	$\in$ **P** (Theorem 5.1 in [8])	**NP**-complete (Theorem 5.3 in [8])
CVLH over $\mathcal{S}^{\mathsf{poly}}_{\mathsf{poly}}$	$\in$ **P** (Theorem 5.1 in [8])	$\in$ **QMA** (Theorem 5.4 in [8])
CVLH over $\mathcal{S}^{\mathsf{exp}}_{r}$	$\in$ **P** (Theorem 5.1 in [8])	$\in$ **NTIME**$(n^{O(r)})$ (Theorem 5.5 in [8])
CVLH	$\in$ **P** (Theorem 5.1 in [8])	**RE**-hard (Theorem 5.6 in [8])

3 Details of the contributions

In what follows, we detail our contributions and provide some intuition. All definitions and results are stated informally and we refer to [8] for formal statements and proofs.

Gaussian computations

Let us first consider the power of bounded-error quantum computations using polynomial Hamiltonians of degree at most 2. The unitary gates generated by such Hamiltonians are known as Gaussian gates. It is well-known that Gaussian gates are efficiently simulatable in polynomial time **P** when acting on Gaussian states [4], i.e., states that may be obtained from the vacuum using Gaussian gates [17], together with Gaussian measurements, i.e., projection onto Gaussian states. We define a model of Gaussian computations as follows (see Definition 3.1 of [8] for a formal statement):

▶ **Definition 1** (Gaussian dynamical computations, informal)**.** *Gaussian dynamical computation (***GDC***) is the class of problems that can be solved by evolving input Gaussian states via logspace uniform quadratic Hamiltonians for polynomial time, followed by measuring a single mode in the position basis (see Section 2.3.3 of [8] for more details about the formalism). The computation accepts if the measured outcome has a value greater than a fixed constant b and rejects if it is below a fixed constant a.*

GDC allows evolving a quantum state according to different Gaussian Hamiltonians one after the other, so long as the total time and number of Hamiltonians do not exceed a polynomial bound. Gaussian computations are known to be continuous-variable analogs

of the so-called Clifford computations in the discrete-variable case. Clifford computations are also known to be classically simulatable in polynomial time, by the Gottesmann–Knill theorem [19]. In [2], Aaronson and Gottesman showed that one can actually simulate Clifford computations (i.e., sample from one qubit) in the complexity class $\oplus\mathbf{L}$, which is believed to be strictly contained in $\mathbf{P}$. Performing linear algebra (such as computing the determinant) over $\mathbb{Z}_2$ is a complete problem for this class. It is natural to ask whether a continuous-variable analog of this result holds. Our first result resolves this question in the affirmative (see Theorem 3.1 in [8] for a formal statement):

▶ **Theorem 2** (The computational power of Gaussian dynamics, informal)**.** *The power of Gaussian computations up to logspace reductions is captured by bounded-error quantum logspace (**BQL**) and, equivalently, the problem of inverting a well-conditioned matrix.*

The proof is based on the symplectic formulation of Gaussian operators [17].

Recently and independently, in [3], it was shown that simulating a particular class of gate-based Gaussian computations over exponentially many modes is **BQP**-complete. Recall that approximate matrix inversion is a **BQP**-complete problem when the matrix under consideration is sparse and well-conditioned [14]. Although our models and results are technically different, both reveal a strong connection between Gaussian computations and linear algebra.

Bounded-error continuous-variable quantum polynomial time (CVBQP)

Arguably, there are many ways one can define a continuous-variable version of **BQP** based on how computation is being performed. We consider bosonic computations using gates generated by polynomials of constant degree in the position and momentum bosonic operators, which are ubiquitous in quantum physics, with particle-number measurements. We define gate set-dependent classes (see e.g., Definition 3 below) and investigate the computational power of these models for different choices of gate sets. In particular, we show that bosonic computations based on specific gates generated by degree-4 and 2-local Hamiltonians can perform universal (discrete-variable) quantum computation on an *input vacuum state and without requiring feed-forward of measurement outcomes* (see Theorem 4.1 of [8]). Interestingly, the equivalence between these different continuous-variable classes is unknown, in part due to the delicate features of unbounded operators such as those pointed out in the introduction. Some of these relationships, such as the fast compiling of polynomial-degree Hamiltonians into Gaussian and cubic phase gates, are outstanding open questions; see, e.g., [40, 27].

We then consider bosonic computations based on a family of circuits generated by cubic phase gates e^{iX^3} and Gaussian gates e^{iH}, where H is a quadratic Hamiltonian in X and P. Notably, this gate set is believed/conjectured to be universal for the set of unitaries generated by arbitrary polynomials over an arbitrary number of modes [40, 27], a claim that was proven in a controllability sense in [45]. Due to [20], this computational model, when equipped with the ability of performing Gaussian measurements and feed-forward, is capable of performing universal (discrete-variable) quantum computation, when complex input states known as Gottesman–Kitaev–Preskill states are available.

Gaussian gates are closed under multiplication, which is one way we can understand the classical simulation of Gaussians. The single-mode Gaussian dynamics can furthermore be understood via specific integrable classical equations of motion (known as Calogero–Moser dynamics) [10]. However, once we add higher-degree gates to the gate set, the operators generated by the resulting gates generate a vastly larger set of operators, and the single-mode

dynamics becomes chaotic [33, 45]. Define $\mathbf{CVBQP}[X^3]$ as the class of decision problems that are solvable using quantum circuits based on Gaussian and cubic phase gate sets (see Definition 4.4 of [8] for a formal definition):

▶ **Definition 3** ($\mathbf{CVBQP}[X^3]$, informal)**.** $\mathbf{CVBQP}[X^3]$ *is the class of decision problems that are solvable with a bounded probability of error by applying a polynomial-time uniform sequence of Gaussian and cubic phase gates to the vacuum state and measuring the number of particles at the end, with the promise that the energy (average particle number) at the output is polynomially bounded.*

We show that, in between the computation, the energy of states prepared from the vacuum by polynomial-time sequences of Gaussian and cubic phase gates is upper bounded by a doubly-exponential function of the number of gates and modes (see Proposition 4.3 of [8]). The reason for such drastic energy growth in the system is that consecutive application of cubic and Gaussian gates can lead to repeated squaring of basic observables. This, in turn, leads to an upper bound of **EEXP** on the strong simulation (computing output amplitudes up to exponential precision) of $\mathbf{CVBQP}[X^3]$ when the final measurement is made in the computational basis. We then apply standard depth reduction techniques to bring the complexity down to **EXPSPACE** (see Theorem 4.2 of [8] for a formal statement):

▶ **Theorem 4** (Upper bound on the computational power of Gaussian and cubic phase gates, informal)**.** *Bosonic computations consisting of Gaussian and cubic phase gates on input vacuum and measurement in the Fock basis can be strongly simulated in* **EXPSPACE**.

This theorem assumes no energy upper bound for each of the computational steps. When introducing an energy (average particle number) upper bound of E^* in the above result, we note that the space complexity of the classical simulation in the single-mode case scales as $O(\mathsf{log}(E^*))$.

Next, we focus on the problem of computing expectation values at the output of $\mathbf{CVBQP}[X^3]$ circuits for a low-degree observable O (see Definition 4.7 and Theorem 4.7 of [8] for formal statements):

▶ **Theorem 5** (The computational complexity of bosonic expectation values, informal)**.** *The problem of computing expectation values of low-degree observables for states prepared by applying Gaussian and cubic phase gates to the vacuum can be solved in* **PSPACE**.

Proof sketch. Let $W_i := U_i \cdots U_1$ where the unitary gates $\{U_i\}_{1\leq i\leq n}$ selected from the Gaussian and cubic-phase gateset, are sequentially applied to a vacuum state. Suppose that $W_i X_1 W_i^\dagger$ takes the form $O_i = \sum_{\mu,\nu} \alpha_{\mu,\nu} X^\mu P^\nu$. Here we used the multi-index notations $\mu = (\mu_1, \cdots, \mu_n) \in \mathbb{Z}_+^n$ and $X^\mu = X_1^{\mu_1} \cdots X_n^{\mu_n}$. Let $O_{i+1} = U_{i+1} O_i U_{i+1}^\dagger$ be the value after the application of the next gate. We can write $O_{i+1} = \sum_{\mu,\nu} \alpha_{\mu,\nu} (U_i X U_i^\dagger)^\mu (U_i P U_i^\dagger)^\nu$. If U_{i+1} is a Gaussian gate, then the degree of X and P in O_i will be the same as that of O_{i+1}. However, if U_{i+1} is a cubic phase gate, then the degree of O_{i+1} is at most twice the degree of O_i. That is because of the unitary evolution due to the cubic-phase gate $U_i P U_i^\dagger = P + c \cdot X^2$, where c is a constant. As a result, the degree of O_T is at most 2^T. We note that coefficients involving the expansion of O_T into a normal form may be doubly exponentially large. Hence, the naive brute-force approach runs in exponential space. To bring the complexity upper bound down to **PSPACE**, we give a polynomial-time *parallel algorithm* using exponentially many processors. Standard results in computational complexity imply a **PSPACE** upper bound (see, for instance, [43]). We further show that the upper bound remains true for the multimode case, using the fact that the only multimode operators we need to add are the so-called two-mode SUM gates. ◀

We emphasize that the standard relationships outlined in Figure 1 indicate a **PP** upper bound on **BQP**. Proving a **EXPSPACE** upper bound on **CVBQP**$[X^3]$ (outlined in Theorem 4) already utilizes nontrivial tools, and we do not know if a stronger upper bound such as **PP** would hold in this case based on current techniques. Indeed, from the previous proof sketch, a remarkable feature of the cubic phase gate becomes apparent: starting with a single position operator X and applying T cubic phase gates interleaved with suitable Gaussian gates (the Fourier gate $e^{i\frac{\pi}{4}(X^2+P^2)}$, mapping X to P and P to $-X$), one can perform repeated squaring, i.e., obtain an observable of the form $\propto X^{2^T} + \cdots$ after T rounds [1]. This implies that doubly exponentially large numbers may naturally arise after polynomially many gates in the continuous-variable setting, in stark contrast with the discrete-variable setting.

The boundedness problem

Next, we aim to study the complexity of estimating the ground state energy of a bosonic Hamiltonian. Due to the infinite-dimensional setting, however, the spectrum may be unbounded. Hence, we first formulate and study the problem of deciding whether a bosonic Hamiltonian has a bounded ground state energy (see Definition 5.3 of [8] for a formal definition). Note that this problem is equivalent to deciding whether a bosonic Hamiltonian has a bounded spectrum, by checking boundedness of the ground energy for H and $-H$. The spectrum of any Hamiltonian that is a polynomial of odd degree (such as X or X^3 or $X^2P + PX^2$, etc.) is not bounded, as can be seen by computing the expectation value for an arbitrary coherent state (eigenstates of the operator $X + iP$), so we focus on polynomial Hamiltonians of even degree.

It turns out that the quadratic (Gaussian) case is solvable in polynomial time via a reduction to the problem of deciding whether a polynomial-size matrix, which may be computed efficiently from the coefficients of the Hamiltonian, is positive semi-definite (see Theorem 5.1 of [8]).

In the non-Gaussian case, we prove that the problem is significantly harder, even for degree-4 Hamiltonians (see Theorem 5.2 of [8] for a formal statement):

▶ **Theorem 6** (Complexity of the boundedness problem, informal)**.** *The problem of deciding whether the spectrum of a bosonic Hamiltonian with degree* 4 *is bounded is* **co-NP***-hard .*

Proof sketch. The proof proceeds via reduction from matrix copositivity, which is the problem of deciding, given $M \in \mathbb{R}^{n\times n}$, whether $\langle x| M |x\rangle \geq 0$ for all $|x\rangle \in \mathbb{R}^n_{\geq 0}$ with non-negative entries. This problem is known to be **co-NP**-complete [36]. ◀

How hard is it to find a witness for the boundedness of the ground state energy? From the above, if we could find it in polynomial time in general, then we would at least collapse **co-NP** to **P**, which is highly unlikely. However, we may find a procedure to achieve this goal at least in some instances. This is what we do next, based on a sum-of-squares technique.

Consider a polynomial Hamiltonian in the form $H = \sum_{\mu,\nu\in\mathbb{Z}^n_+} \frac{1}{2} h_{\boldsymbol{\mu},\boldsymbol{\nu}} \{X^{\boldsymbol{\mu}}, P^{\boldsymbol{\nu}}\}$, where we used multi-index notation (in Lemma 5.2 of [8] we show that any polynomial Hamiltonian can be brought to this form with real coefficients $h_{\boldsymbol{\mu},\boldsymbol{\nu}} \in \mathbb{R}$). The classical polynomial $p_H : \mathbb{R}^{2n} \to \mathbb{R}$ corresponding to H is defined as

$$p_H(x_1, \cdots, x_n, p_1, \cdots, p_n) = \sum_{\boldsymbol{\mu},\boldsymbol{\nu}\in\mathbb{Z}^n_+} h_{\boldsymbol{\mu},\boldsymbol{\nu}} x^{\boldsymbol{\mu}} p^{\boldsymbol{\nu}}.$$

[1] We thank Francesco Arzani for pointing out this fact.

We prove the following (see Proposition 5.4 of [8] for a formal statement):

▶ **Proposition 7** (Checking boundedness for degree-4 Hamiltonians, informal). *Let H be a bosonic Hamiltonian of degree 4. If p_H is a sum-of-squares polynomial, then the spectrum of H is bounded from below by an efficiently computable constant.*

As a result, a sum-of-squares approach provides a sound algorithm for deciding boundedness of the ground state energy of polynomial Hamiltonians in the $d = 4$ case, meaning that if we find a valid sum-of-squares decomposition for p_H, then H is bounded, but if p_H is not a sum-of-squares that does not imply unboundedness for H. Since the degree of p_H is constant, one can look for a sum-of-squares decomposition by running a polynomial-time semi-definite program [38]. Note that if p_H is not a sum of squares and we conjugate H by a Gaussian unitary U (which does not change the degree of H) and try again, we may find a different valid witness for boundedness.

The continuous-variable local Hamiltonian problem

We consider the case of general polynomial Hamiltonians and define the continuous-variable local Hamiltonian problem as follows (see Definition 5.2 of [8] for a formal definition):

▶ **Definition 8** (The continuous-variable local Hamiltonian problem, informal). *Let $\mathcal{S} \subset \mathcal{H}$ be a subset of continuous-variable quantum states. The continuous-variable local Hamiltonian problem $\mathsf{CVLH}^d_{\mathcal{S}}$ is the problem of estimating the lowest energy of a poynomial Hamiltonian H of degree d over the set $\mathcal{S}$.*

Note that the name *local* for this problem comes here from the fact that any polynomial Hamiltonian of degree d is at most d-local by definition.

For Gaussian Hamiltonians, our solution to the boundedness problem also provides a polynomial-time algorithm to estimate the ground state energy, thus placing the continuous-variable local Hamiltonian problem $\mathsf{CVLH}^2_{\mathcal{H}}$ for Gaussian Hamiltonians in **P** (see Theorem 5.1 of [8]).

For non-Gaussian Hamiltonians, in the case of states with bounded particle number, a result of [11] proves that the problem of estimating the ground state energy of the Bose–Hubbard model at finite (polynomial) number of bosons is **QMA**-complete. This Hamiltonian is of the form $H_{\mathrm{bh}} = t \cdot \sum_{i,j} A_{i,j}\, a_i^\dagger a_j + J \cdot \sum_i N_i(N_i - 1)$, where $N = (X^2 + P^2 - 1)/2$ and $a = (X + iP)/\sqrt{2}$, and where $A_{i,j}$ is the (i,j)-th entry of the adjacency matrix of an undirected graph. Note that the Hamiltonian H_{bh} conserves the number of particles, and hence, this operator may be thought of as a finite-dimensional Hamiltonian. In particular, denoting by $\mathcal{H}_n$ the set of states with less than n particles, this shows that $\mathsf{CVLH}^d_{\mathcal{H}_n}$ is **QMA**-hard already for $d = 4$ (and thus $\mathsf{CVLH}^4_{\mathcal{H}}$ as well).

In general, the complexity of the problem for non-Gaussian Hamiltonians depends significantly on the complexity of the set of states one optimizes. Following [9, 10], we consider the stellar rank of a quantum state as a parameter for specifying this set of states (see Section 2.3 of [8] for a brief review of the stellar rank). In short, to any continuous-variable quantum state $|\psi\rangle$ over n modes, one can associate a holomorphic function $F^\star_\psi : \mathbb{C}^n \to \mathbb{C}$. When this holomorphic function can be decomposed as a product of a polynomial P and a Gaussian G, i.e., $F^\star_\psi(z_1, \cdots, z_n) = P(z_1, \cdots, z_n)G(z_1, \cdots, z_n)$, the degree of the polynomial P defines the stellar rank r of $|\psi\rangle$. Otherwise, the stellar rank is infinite. When $r = 0$, we obtain the set of all Gaussian states, which can be produced by Gaussian gates applied to the vacuum state. The stellar rank can be finite (e.g., the stellar rank of n indistinguishable

particles is n). It can also be infinite, e.g., $\alpha_1 |\psi_1\rangle + \alpha_2 |\psi_2\rangle$ for Gaussian states $|\psi_1\rangle, |\psi_2\rangle$ ($\alpha_1, \alpha_2 \in \mathbb{C}$). We further constrain the energy of these states with respect to the number operator N (see Definition 5.5 of [8] for a formal definition).

For zero stellar rank, we show (see Theorem 5.3 of [8] for a formal statement):

▶ **Theorem 9** (Lowest energy over Gaussian states, informal)**.** *The problem of estimating the lowest energy of an n-mode polynomial bosonic Hamiltonian of constant degree over the set $\mathcal{S}_0^{\mathsf{exp}}$ of states of stellar rank 0 (Gaussian states) with energy (average particle number) at most $\mathsf{exp}(n)$ is **NP**-complete.*

Proof sketch. **NP**-hardness comes from a reduction from matrix non-copositivity (rather than matrix copositivity for Theorem 6). To place the problem in **NP**, note that the expectation value of a constant-degree polynomial Hamiltonian over a Gaussian state with energy at most $\mathsf{exp}(n)$ may be computed efficiently, so the **NP** witness is a description of the Gaussian state of lowest energy. ◀

For logarithmically and polynomially-bounded stellar ranks, we show (see Theorem 5.4 of [8] for a formal statement):

▶ **Theorem 10** (Lowest energy over bounded stellar rank, informal)**.** *The problem of estimating the lowest energy of an n-mode polynomial bosonic Hamiltonian of constant degree over the set $\mathcal{S}_{\mathsf{poly}}^{\mathsf{poly}}$ of states of stellar rank $r = \mathsf{poly}(n)$ with energy (average particle number) at most $\mathsf{poly}(n)$ is in **QMA** (this result holds up to Conjecture 2 of [8]).*

Proof sketch. To place the problem in **QMA**, we use the fact that any state of finite stellar rank is related by a Gaussian unitary G to a state of bounded particle number $|c\rangle$ [10]. This allows us to efficiently rewrite the optimisation in terms of a (sparse) finite-dimensional Hamiltonian, once the Gaussian unitary corresponding to the lowest energy state is known. The **QMA** witness is then given by a classical description of that Gaussian unitary G provided in the computational basis, together with a finite-dimensional state $|c\rangle$ that is the ground state of the finite-dimensional Hamiltonian, such that $G|c\rangle$ is the lowest energy state of the original Hamiltonian.

When the stellar rank is logarithmically bounded instead, $|c\rangle$ has an efficient classical description and the witness can be made fully classical. ◀

Finally, we consider the general case of unbounded stellar rank, with no restrictions on the set of states over which the optimisation takes place, and we show (see Theorem 5.6 of [8] for a formal statement):

▶ **Theorem 11** (The complexity of the continuous-variable local Hamiltonian problem, informal)**.** *The problem of estimating the ground energy of a polynomial bosonic Hamiltonian of constant degree is undecidable. This problem is already undecidable for $d = 8$.*

Proof sketch. The proof of this result proceeds via a reduction from Hilbert's tenth problem (a Nüllstellensatz problem over the integers) due to [30], combined with the fact that there exist explicit undecidable polynomial equations over non-negative integers of degree 4 and a constant number of unknowns [25]. ◀

Continous-Variable Quantum Merlin Arthur games

Finally, we introduce a continuous-variable analog of **QMA** (see Definition 5.1 of [8] for a formal definition):

▶ **Definition 12** (Continuous-variable quantum Merlin-Arthur, informal)**.** **CVQMA** *is the class of decision problems, for which a solution encoded in a continuous-variable quantum state can be verified efficiently by a* **CVBQP** *machine.*

Motivated by the relationship between the local Hamiltonian problem and **QMA** in the discrete-variable case based on Kitaev's history state construction [28], we aim to connect the class **CVQMA** to the continuous-variable local Hamiltonian problem. We give the basis for a continuous-variable history state construction providing such a connection (see Section 5.3 of [8]), and identify the challenges associated with such a construction.

4 Outlook

The study of continuous-variable quantum computations may have various interactions with the foundations of computational complexity, computability, and quantum mechanics, which we discuss in the next section. After that, in Section 4.1.1, we list several open questions.

4.1 Discussion

Energy of continuous-variable computations

One of the key insights of this work is that by alternating Gaussian and cubic phase gates, the average energy of the system – even for a single mode – can grow to be doubly exponential in the number of cubic phase gates used in the circuit. This observation suggests that strong simulation (calculating each amplitude individually) of continuous-variable quantum systems could be significantly more challenging than for discrete-variable ones, potentially being hard for complexity classes like **PSPACE** or even **EXPSPACE**. This suggests that on top of time and space complexity, energy plays a significant role in the computational power of bosonic systems.

In practical physical experiments, the energy must be supplied by the experimentalist, requiring a physical definition of computational cost that accounts for time, space, and energy. Specifically, if one is willing to expend up to E^* units of energy, our results show that a single mode can be simulated within $\mathbf{SPACE}(O(\log E^*))$. This highlights a trade-off between time, space, and energy, which we believe deserves a thorough examination. Understanding time, energy, and space tradeoffs for multimode systems equipped with multi-mode non-Gaussian gates is an interesting open question.

In our definition of **CVBQP**, we enforce the "promise" that the quantum state in the beginning and the end has limited energy, as measured by the average particle number, but it may take any value in between. A natural question to ask is: is the power of this model the same as the one where we impose restrictions on average energy at any point during the computation? See Figure 3 for a visual depiction. We can ask a similar question about classical continuous (time and/or space) models of computation where we are promised that the system's state is effectively discrete (e.g., a two-level system) at the beginning and the end. Still, the system can utilize its continuous degrees of freedom in between to an arbitrary precision. Is the power of this hypothetical model the same as digital computation? We first note that without such promise in the beginning, it is not clear how one can program such a system, and without the promise on the energy in the end, no physically viable device would be able to measure the quantum state. Even if we assume the existence of such a hypothetical device, it is not difficult to approximately predict the output of such a hypothetical device (we output random numbers because, at high energy, no concentration of measure is expected).

We may face the objection that assuming no mechanism to bound energy (e.g., dissipation), this promise is unreasonable because we cannot verify it. We first note that it is not difficult to come up with trivial examples that satisfy such a promise. Consider the unitary $UU^\dagger$ where $U|0\rangle$ has high energy. Clearly, $UU^\dagger$ has bounded energy at the beginning and the end and very high energy in between. Second, promises that are possibly difficult to verify are common in quantum complexity theory, e.g., promise on the gap of a Hamiltonian or promise that a **BQP** computation either accepts with probability $\geq 2/3$ or $\leq 1/3$. We have a similar scenario for the energy promise.

As a thought experiment, assume we have a fragile quantum processing device that breaks if it holds more than N_0 particles. Now, we design a quantum experiment such that the system (on average) has $\ll N_0$ particles at the beginning and the end but may (on average) have $\gg N_0$ particles in a way that many of the computational paths involve $\ll N_0$ particles and many involve $\gg N_0$. In the end, we measure the device's output and measure $\ll N_0$ particles. Do we measure the device to be broken or unbroken? This is similar to the Schrödinger's cat (or Wigner's) paradox. In the mentioned thought experiment, the device played the cat's role. In other words, if the device's condition (i.e., broken or unbroken) is determined only at the time of measurement, then **CVBQP** with mild (or no) energy restriction in between might be a plausible model. Otherwise, if broken paths are forbidden, then **CVBQP** with energy bound on the entire computation path is a more reasonable model from a physical standpoint. We note that in an actual experiment, energy is pumped from an outside source, and the closedness of the experiment is only an approximation.

Connections with the extended Church–Turing thesis

Our result leaves open the possibility that **CVBQP** $\not\subseteq$ **BQP**; due to doubly exponential growth of energy in between computations, such a separation is plausible. What would that imply about the nature of computation in the physical world? If **CVBQP** is a plausible model for computation in the physical world, then **CVBQP** $\not\subseteq$ **BQP** seems to imply a contradiction to the *extended Church Turing Thesis*. But how realistic is **CVBQP** as a model of computation? In particular, how should one determine the energy cap in between the computations? Consider scattering amplitude for quantum field theories, where the specific amplitude sets the number of particles at the beginning and the end, and still, fluctuations in the vacuum may lead to many particle creations, and annihilations can occur in between. Are bounds on the fluctuations of the vacuum (see, for instance, [18]) such that the computational complexity of scattering amplitudes do not exceed that of **BQP**? Are there computational phase transitions depending on γ_1 and γ_2?

What can we say about noisy systems? Suppose we define **DissCVBQP**(γ_1, γ_2) as a dissipated model where we have a mechanism that pumps bosons into the system with rate γ_1 and another mechanism that bosons are emitted to the environment with rate γ_2. How does the power computational complexity of **DissCVBQP**(γ_1, γ_2) depend on γ_1 and γ_2? Can we show for physically relevant parameters the power of this model is **BQP**-complete?

Logarithmic number of cubic phase gates

Standard results in the discrete-variable model of quantum computing (e.g., [7]) imply that starting with a computational basis, discrete-variable quantum circuits with a logarithmic number of T gates and polynomially many Clifford gates can be efficiently strongly simulated on classical computers. Can we prove a similar result for continuous-variable systems, i.e., can we show that starting from the vacuum state, a polynomial-size quantum circuit with a

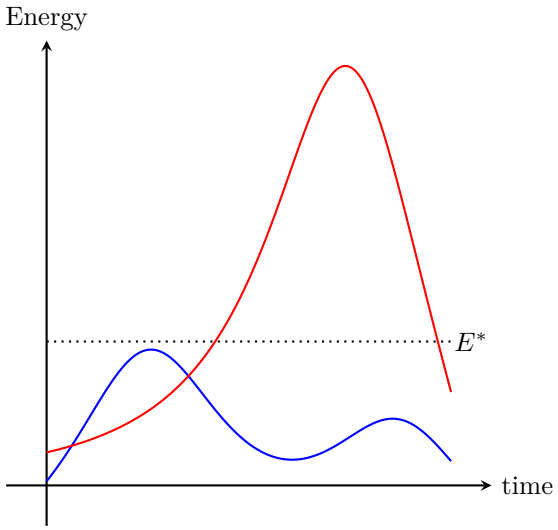


Figure 3 Energy bound conditions for quantum computations. The red curve depicts a system that is promised to have low energy at the beginning and the end of the computation, but may have arbitrary high energy in between. The blue curve depicts a model where we impose energy restrictions at any point during the computation.

logarithmic number of cubic phase gates can be simulated on a classical computer? Based on our results, the best upper bound we could prove for this problem was **PSPACE**. We furthermore note that, due to repeated squaring of energy, the "effective dimension" that the quantum system explores is exponential even if the circuit has, at most, a logarithmic number of cubic phase gates. Based on this intuition, we conjecture that (noiseless) continuous variable quantum circuits with a logarithmic number of cubic phase gates cannot be simulated efficiently on a classical computer.

Noiseless model as a foundation to study noisy systems

Even though a realistic model for the experimental implementation of bosonic computations is a dissipative (or noisy) model, noise models are various, and by understanding the computability limits of the noiseless system, one can gain a reliable foundation to study computational complexity after imposing different restrictions on the model. A similar approach has been pursued for computational complexity in the discrete-variable domain where we define **BQP** as an idealized noiseless quantum computing model and pin down its computational complexity. We then use this foundation to study the model's different variants and restrictions. For instance, we know that computing the amplitudes of a noiseless quantum circuit is #**P**-complete. Obviously, this result is not directly relevant to understanding the cost of simulating practical problems. However, this observation has been utilized via tools in computational complexity to lay the foundations for demonstrating quantum speedup in noisy or restricted models such as Boson Sampling [1]. Moreover, we describe a way to encode **BQP** computations in bosonic subspaces (see Theorem 4.1 of [8]) relying on the Solovay–Kitaev theorem [13], but this result does not appear to be robust to noise, because the noise may induce leakage to unbounded regions of state space where the theorem no longer holds. This motivates further study of robust quantum computations in continuous-variable systems.

Truncating CV systems using the stellar rank

Standard approaches to describe continuous-variable quantum systems using a finite number of variables include (i) truncating their infinite-dimensional Hilbert space, which amounts to restricting to bounded particle-number supports, or (ii) keeping track only of the covariance matrix and displacement vector of the state. These have immediate shortcomings: (i) is not stable under Gaussian evolutions, which may be thought of as computationally easy [4], and truncating a Gaussian state usually makes it non-Gaussian; (ii) only faithfully describes Gaussian computations.

Our results suggest that the stellar rank may be a meaningful approach for describing continuous-variable quantum systems, in particular for studying ground state problems for continuous-variable systems [41]. Informally, the stellar rank combines both approaches (i) and (ii), as states of finite stellar rank can always be expressed as (mixture of) Gaussian unitary operators acting on (core) states of finite support.

4.1.1 Open questions

Since the aim of the present work is to lay foundations for a theory of bosonic quantum complexity, it naturally leads to many open questions, some of which we list in the following:

1. The most immediate open question from this work is whether we can bring the **EXPSPACE** upper bound on **CVBQP** to smaller complexity classes such as **PSPACE** or even **PP**. What about lower bounds? Due to the doubly exponentially large dimensionality of the effective Hilbert space, it is natural to conjecture that strong simulation of **CVBQP** is hard for a complexity class such as **PSPACE** (or even **EXPSPACE**) which is believed to be strictly larger than **PP**. Such a result would be important evidence that **CVBQP** may surpass the power of **BQP**.
2. A related question is: what energy bound on **CVBQP** makes it equal to **BQP**? In an actual physical computation (such as a bosonic system subject to dissipation), one would expect that the energy stays polynomially bounded. Is it the case that **CVBQP** = **BQP** under the promise that the energy stays polynomially bounded throughout the computation? Can we prove that a variant of **CVBQP** subject to dissipation is equivalent to **BQP**?
3. It is usually assumed in the continuous-variable quantum information literature that a single non-Gaussian gate together with all Gaussian ones is sufficient to perform "universal" quantum computations. However, this notion of universality is somewhat restricted, as it relates to the ability to approximate evolutions generated by polynomial Hamiltonians [33]. In particular, is the cubic phase gate (for instance) and Gaussian gate set *universal*, in the sense that *any state* can be reached to arbitrary precision from the vacuum state using unitary gates from this set? See [45] for a formal statement of this open problem, and [35] for an example of a continuous-variable gate set satisfying this property.
4. In the discrete-variable setting, the computational power of quantum circuits is essentially independent of the choice of universal gate set. Is it also the case in the continuous-variable setting? If so, can a continuous-variable Solovay–Kitaev theorem [31, 13] be derived for these gates? See [5] for such a result in the case of Gaussian gates. This also relates to the existence of fast compilation algorithms for bosonic gates [40, 26, 27].
5. What is the precise complexity of the Hamiltonian spectrum boundedness problem? We prove **co-NP**-hardness and conjecture hardness for **co-QMA**. Can we show **co-QMA** is an upper bound? We describe a sound algorithm for verifying boundedness in the $d = 4$ case. Do similar results exist for $d > 4$?

6. After possible conjugations with arbitrary unitary matrices, can we write a Hamiltonian as a sum-of-squares of other Hamiltonians (c.f. [23])? We have obtained partial results in this direction for degree-4 polynomial bosonic Hamiltonians.
7. We can define families of Hamiltonians with a ground state of a given exact or approximate stellar rank r. Is there a procedure to decide the opposite? I.e., given the description of the Hamiltonian and an approximation parameter, decide whether it has an approximate ground state of stellar rank $< r$.
8. Our results indicate if we minimize the energy of a continuous-variable Hamiltonian over an ensemble with polynomially bounded energy (particle number) and stellar rank r, then for $r = \mathsf{poly}(n)$ it is contained in **QMA**. Can we show that these containments are tight? What about $r = \mathsf{exp}(\mathsf{n})$ and higher particle numbers? Proving such results would involve continuous-variable versions of Kitaev's history state [29, 28] in the continuous-variable setting. This construction would involve nontrivial ideas, which we leave for future work.
9. What is the complexity of the continuous-variable local Hamiltonian problem for other natural families of continuous-variable quantum states? For instance, the family of quantum states that are superpositions of $\mathsf{poly}(n)$ Gaussians? This class of states has recently been considered in the context of classical simulation of bosonic computations, leading to the introduction of the continuous-variable notion of *Gaussian rank* [15, 22], akin to the discrete-variable notion of *stabilizer rank* [7]. Note that the stellar rank for such states is typically infinite.
10. What is the complexity of deciding whether the spectrum of a polynomial bosonic Hamiltonian has a continuous part or if is it discrete? When it is discrete, is CVLH **RE**-complete?

Bosonic quantum computations also provide a natural setting for generalizing classical complexity theory over the reals (e.g., in the Blum–Shub–Smale model [6]) to the quantum case, given that bosonic Hamiltonians may have a continuous spectrum. We expect that most of our results should have counterparts over the reals.

References

1 Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 333–342, 2011. doi:10.1145/1993636.1993682.

2 Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, 2004. doi:10.1103/PhysRevA.70.052328.

3 Alice Barthe, M Cerezo, Andrew T Sornborger, Martin Larocca, and Diego García-Martín. Gate-based quantum simulation of Gaussian bosonic circuits on exponentially many modes. *arXiv preprint*, 2024. doi:10.48550/arXiv.2407.06290.

4 Stephen D Bartlett, Barry C Sanders, Samuel L Braunstein, and Kae Nemoto. Efficient classical simulation of continuous variable quantum information processes. *Physical Review Letters*, 88(9):097904, 2002. doi:10.1103/PhysRevLett.88.097904.

5 Simon Becker, Nilanjana Datta, Ludovico Lami, and Cambyse Rouzé. Energy-constrained discrimination of unitaries, quantum speed limits, and a Gaussian Solovay-Kitaev theorem. *Physical Review Letters*, 126(19):190504, 2021. doi:10.1103/PhysRevLett.126.190504.

6 Lenore Blum, Felipe Cucker, Mike Shub, and Steve Smale. *Complexity and real computation*. Springer Science & Business Media, 1998. doi:10.1007/978-1-4612-0701-6.

7 Sergey Bravyi, Graeme Smith, and John A Smolin. Trading classical and quantum computational resources. *Physical Review X*, 6(2):021043, 2016. doi:10.1103/PhysRevX.6.021043.

8 Ulysse Chabaud, Michael Joseph, Saeed Mehraban, and Arsalan Motamedi. Bosonic quantum computational complexity. *arXiv preprint*, 2024. doi:10.48550/arXiv.2410.04274.

9 Ulysse Chabaud, Damian Markham, and Frédéric Grosshans. Stellar representation of non-Gaussian quantum states. *Physical Review Letters*, 124(6):063605, 2020. doi:10.1103/PhysRevLett.124.063605.

10 Ulysse Chabaud and Saeed Mehraban. Holomorphic representation of quantum computations. *Quantum*, 6:831, 2022. doi:10.22331/Q-2022-10-06-831.

11 Andrew M Childs, David Gosset, and Zak Webb. The Bose-Hubbard model is QMA-complete. In *Automata, Languages, and Programming: 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I 41*, pages 308–319. Springer, 2014. doi:10.1007/978-3-662-43948-7_26.

12 Carsten Damm. Problems complete for ⊕L. In *Aspects and Prospects of Theoretical Computer Science: 6th International Meeting of Young Computer Scientists Smolenice, Czechoslovakia, November 19–23, 1990 Proceedings 6*, pages 130–137. Springer, 1990. doi:10.1016/0020-0190(90)90150-V.

13 Christopher M Dawson and Michael A Nielsen. The Solovay-Kitaev algorithm. *arXiv preprint quant-ph/0505030*, 2005. doi:10.48550/arXiv.quant-ph/0505030.

14 Danial Dervovic, Mark Herbster, Peter Mountney, Simone Severini, Naïri Usher, and Leonard Wossnig. Quantum linear systems algorithms: a primer. *arXiv preprint*, 2018. doi:10.48550/arXiv.1802.08227.

15 Beatriz Dias and Robert Koenig. Classical simulation of non-Gaussian bosonic circuits. *arXiv preprint*, 2024. doi:10.48550/arXiv.2403.1905.

16 Bill Fefferman and Zachary Remscrim. Eliminating intermediate measurements in space-bounded quantum computation. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1343–1356, 2021. doi:10.1145/3406325.3451051.

17 Alessandro Ferraro, Stefano Olivares, and Matteo GA Paris. Gaussian states in continuous variable quantum information. *arXiv preprint*, 2005. doi:10.48550/arXiv.quant-ph/0503237.

18 LH Ford. Negative energy densities in quantum field theory. *International Journal of Modern Physics A*, 25(11):2355–2363, 2010. doi:10.1142/S0217751X10049633.

19 Daniel Gottesman. The Heisenberg representation of quantum computers. *arXiv preprint*, 1998. doi:10.48550/arXiv.quant-ph/9807006.

20 Daniel Gottesman, Alexei Kitaev, and John Preskill. Encoding a qubit in an oscillator. *Physical Review A*, 64(1):012310, 2001. doi:10.1103/PhysRevA.64.012310.

21 Arne L Grimsmo and Shruti Puri. Quantum error correction with the Gottesman-Kitaev-Preskill code. *PRX Quantum*, 2(2):020101, 2021. doi:10.1103/PRXQuantum.2.020101.

22 Oliver Hahn, Ryuji Takagi, Giulia Ferrini, and Hayata Yamasaki. Classical simulation and quantum resource theory of non-Gaussian optics. *arXiv preprint*, 2024. doi:10.48550/arXiv.2404.07115.

23 Matthew B Hastings. Perturbation theory and the sum of squares. *arXiv preprint arXiv:2205.12325*, 2022. doi:10.48550/arXiv.2205.12325.

24 Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. MIP*= RE. *Communications of the ACM*, 64(11):131–138, 2021. doi:10.48550/arXiv.2001.04383.

25 James P Jones. Three universal representations of recursively enumerable sets. *The Journal of Symbolic Logic*, 43(3):549–571, 1978. doi:10.2307/2272832.

26 Timjan Kalajdzievski and Juan Miguel Arrazola. Exact gate decompositions for photonic quantum computing. *Physical Review A*, 99(2):022341, 2019. doi:10.1103/PhysRevA.99.022341.

27 Timjan Kalajdzievski and Nicolás Quesada. Exact and approximate continuous-variable gate decompositions. *Quantum*, 5:394, 2021. doi:10.22331/Q-2021-02-08-394.

28 Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. *FSTTCS 2004. Lecture Notes in Computer Science, vol 3328*, 2004. doi:10.1007/978-3-540-30538-5_31.

29 Julia Kempe and Oded Regev. 3-local hamiltonian is QMA-complete. *arXiv preprint*, 2003. doi:10.48550/arXiv.quant-ph/0302079.

30 Tien D Kieu. Quantum algorithm for Hilbert's tenth problem. *International Journal of Theoretical Physics*, 42:1461–1478, 2003. doi:10.1023/A:1025780028846.

31 A Yu Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191, 1997. doi:10.1070/RM1997v052n06ABEH002155.

32 Emanuel Knill, Raymond Laflamme, and Gerald J Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, 409(6816):46–52, 2001. doi:10.1038/35051009.

33 Seth Lloyd and Samuel L Braunstein. Quantum computation over continuous variables. *Physical Review Letters*, 82(8):1784, 1999. doi:10.1103/PhysRevLett.82.1784.

34 Lars S Madsen, Fabian Laudenbach, Mohsen Falamarzi Askarani, Fabien Rortais, Trevor Vincent, Jacob FF Bulmer, Filippo M Miatto, Leonhard Neuhaus, Lukas G Helt, Matthew J Collins, et al. Quantum computational advantage with a programmable photonic processor. *Nature*, 606(7912):75–81, 2022. doi:10.1038/s41586-022-04725-x.

35 R Vilela Mendes and Vladimir I Man'ko. On the problem of quantum control in infinite dimensions. *Journal of Physics A: Mathematical and Theoretical*, 44(13):135302, 2011. doi:10.1088/1751-8113/44/13/135302.

36 Katta G. Murty and Santosh N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39(2):117–129, 1987. doi:10.1007/BF02592948.

37 Michael A Nielsen and Isaac L Chuang. Quantum computation and quantum information. *Phys. Today*, 54(2):60, 2001. doi:10.1017/CBO9780511976667.

38 Pablo A Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96:293–320, 2003. doi:10.1007/S10107-003-0387-5.

39 Bo Peng, Yuan Su, Daniel Claudino, Karol Kowalski, Guang Hao Low, and Martin Roetteler. Quantum simulation of boson-related hamiltonians: techniques, effective hamiltonian construction, and error analysis. *arXiv preprint*, 2023. doi:10.48550/arXiv.2307.06580.

40 Seckin Sefi and Peter Van Loock. How to decompose arbitrary continuous-variable quantum operations. *Physical review letters*, 107(17):170501, 2011. doi:10.1103/PhysRevLett.107.170501.

41 Paolo Stornati, Antonio Acin, Ulysse Chabaud, Alexandre Dauphin, Valentina Parigi, and Federico Centrone. Variational quantum simulation using non-Gaussian continuous-variable systems. *arXiv preprint*, 2023. doi:10.48550/arXiv.2310.15919.

42 Amnon Ta-Shma. Inverting well conditioned matrices in quantum logspace. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 881–890, 2013. doi:10.1145/2488608.2488720.

43 Jerry L Trahan, Michael C Loui, and Vijaya Ramachandran. Multiplication, division, and shift instructions in parallel random access machines. *Theoretical computer science*, 100(1):1–44, 1992. doi:10.1016/0304-3975(92)90362-J.

44 John Watrous. Quantum computational complexity. *arXiv preprint*, 2008. doi:10.48550/arXiv.0804.3401.

45 Re-Bing Wu, Tzyh-Jong Tarn, and Chun-Wen Li. Smooth controllability of infinite-dimensional quantum-mechanical systems. *Physical Review A*, 73(1):012719, 2006. doi:10.1103/PhysRevA.73.012719.

46 Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, et al. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, 2020. doi:10.1126/science.abe8770.

Directed Hypercube Routing, a Generalized Lehman-Ron Theorem, and Monotonicity Testing

Deeparnab Chakrabarty
Dartmouth College, Hanover, NH, USA

C. Seshadhri
University of California, Santa Cruz, CA, USA

Abstract

Motivated by applications to monotonicity testing, Lehman and Ron (JCTA, 2001) proved the existence of a collection of vertex disjoint paths between comparable sub-level sets in the directed hypercube. The main technical contribution of this paper is a new proof method that yields a generalization of their theorem: we prove the existence of *two* edge-disjoint collections of vertex disjoint paths. Our main conceptual contributions are conjectures on directed hypercube flows with simultaneous vertex and edge capacities of which our generalized Lehman-Ron theorem is a special case. We show that these conjectures imply *directed isoperimetric theorems*, and in particular, the robust directed Talagrand inequality due to Khot, Minzer, and Safra (SIAM J. on Comp, 2018). These isoperimetric inequalities, that relate the directed surface area (of a set in the hypercube) to its distance to monotonicity, have been crucial in obtaining the best monotonicity testers for Boolean functions. We believe our conjectures pave the way towards combinatorial proofs of these directed isoperimetry theorems.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms; Theory of computation → Randomness, geometry and discrete structures

Keywords and phrases Monotonicity testing, isoperimetric inequalities, hypercube routing

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.34

Related Version *arXiv Version*: https://www.arxiv.org/abs/2409.02206

Funding *Deeparnab Chakrabarty*: Supported by NSF-CAREER CCF-2041920 and CCF-2402571.
C. Seshadhri: Supported by NSF CCF-1740850, CCF-1839317, CCF-2402572, and DMS-2023495.

1 Introduction

We let $d \geq 2$ denote a natural number. The directed d-dimensional hypercube graph H has vertices $V(H)$ which correspond to bit-vectors $x \in \{0,1\}^d$, and edges $E(H)$ corresponding to pairs of bit-vectors (x, y) that differ in exactly one coordinate. Edges point from lower Hamming weight vectors to larger ones. We use x_i to denote the ith coordinate of vertex x. There is a natural partial order on the vertices/elements of the Boolean hypercube: $x \preceq y$ iff $\forall i, x_i \leq y_i$. Note that the directed hypercube is precisely the Hasse diagram of this partial order. Equivalently, one can consider the vertices as subsets of $[d]$, and the partial order is given by containment.

Two subsets S, T of $V(H)$ are called a *matched pair* if there exists a *bijection* $\phi : S \to T$ such that $s \prec \phi(s)$ for all $s \in S$; we denote a matched pair by $(S, T; \phi)$. An early writeup of Goldreich-Goldwasser-Ron [30] posed a routing question, inspired by questions in monotonicity testing, which was solved by Lehman and Ron [36]. (More discussion in §4.) They were interested in the following natural question: given any matched pair $(S, T; \phi)$, can one find (edge or vertex) *disjoint* directed paths[1] from S to T? Remarkably, they proved that if all

[1] In their paper, Lehman and Ron consider these paths to be disjoint chains of subsets.

© Deeparnab Chakrabarty and C. Seshadhri;
licensed under Creative Commons License CC-BY 4.0

points in S (resp. T) have the same Hamming weight, then the answer is affirmative: one can find *vertex disjoint paths* from S to T! More precisely, for an integer $0 \leq i \leq d$, let L_i denote the *ith layer* of H, that is, $L_i := \{x \in V(H) \ : \ ||x||_1 = i\}$. We refer to this beautiful statement as the "Lehman-Ron (LR) theorem".

▶ **Theorem 1** (Lehman-Ron Theorem [36]). *Fix any two integers $i < j$. Let $(S, T; \phi)$ be a matched pair with $S \subseteq L_i$ and $T \subseteq L_j$. Then, there are $|S| = |T|$* vertex disjoint paths *between S and T. We refer to such a set of vertex disjoint paths as an* LR solution.

▶ Remark. The paths may not respect the bijection ϕ. More precisely, the above doesn't prescribe vertex disjoint paths from s to $\phi(s)$ for all $s \in S$. There exist concrete counterexamples (one is given in Lehman and Ron's paper attributed to Dan Kleitman) for such paths. A follow-up work by proves that *edge*-disjoint paths from s to $\phi(s)$ don't exist either [16].

We give an alternate proof of this theorem. But more importantly, we use our new proof technique to strengthen the LR theorem. If the terminals are at distance at least 2, there exist *two* edge-disjoint LR solutions.

▶ **Theorem 2.** *Fix any two integers $i < j$ with $j - i \geq 2$. Let $(S, T; \phi)$ be a matched pair with $S \subseteq L_i$ and $T \subseteq L_j$. Then, there are 2 collections of* vertex disjoint paths *between S and T, such that their union is* edge disjoint.

Lehman-Ron's proof of Theorem 1 is by induction on $|S|$ and on the quantity $(j - i)$, the distance between the layers in which S and T lie. The base case of $j - i = 1$ is obvious as the bijection gives us the matching between S and T. The heart of the proof essentially shows the existence of a set of vertices U either in layer L_{j-1} or L_{i+1} and two bijections $\phi' : S \to U$ and $\phi'' : U \to T$ such that $(S, U; \phi')$ and $(U, T; \phi'')$ are matched pairs. This last part is a neat argument which uses Menger's theorem, which is a special case of the max-flow-min-cut theorem, on an auxiliary graph that they create. But how can one get *two* edge disjoint LR solutions? The reader may notice that even the "base case" of $j - i = 2$, that is, when S and T are two levels apart is itself non-trivial (indeed, we don't really know a much simpler way to solve this than the general case). And so, a new idea is needed to prove Theorem 2.

Our proof of the Lehman-Ron theorem brings the *flow-cut duality* idea front and center. We note that Theorem 1 is actually a statement about the structure of flows and cuts in the directed hypercube. More precisely, it states the existence of $|S|$ units of flow from vertices in S to vertices in T when all vertices have *vertex capacity* 1 unit. We exploit the duality between cuts and flows, and more precisely the notion of *complementary slackness*, to give an alternate proof of the LR Theorem. In this flow-cut language, Theorem 2 states the existence of $2|S|$ units of flow when both edges and vertices have capacities (1 and 2 units each, respectively). The existence of the two kinds of capacities makes the argument slightly more involved, but the essence is still the same. For completeness, we show proofs of both Theorem 1 and Theorem 2 in Section 2 and Section 3, respectively.

We end our introduction with a natural conjecture that our techniques have been unable to solve. We discuss the connections between this conjecture and monotonicity testing in Section 4. As the layers L_i and L_j move further apart, there should exist more collections of edge-disjoint LR solutions between S and T.

▶ **Conjecture 3.** *Fix any two integers $i < j$ with $r := j - i$. Let $(S, T; \phi)$ be a matched pair with $S \subseteq L_i$ and $T \subseteq L_j$. Then, there are r collections of* vertex disjoint paths *between S and T, such that their union is* edge disjoint.

Other LR connections. Recent work has generalized the LR theorem different directions in [3]. These results find vertex disjoint paths that "cover" any collection of points specifying certain properties. Consider a subset X of the hypercube that is partitioned into subsets of paths (or chains). Meaning, we partition $X = \bigcup_i X_i$ such that, the vertices of X_i can be ordered according to $\prec$. (Moreover, this is the partition that minimizes the number of sets.) In the vanilla LR setting, each X_i is just $(s, \phi(s))$ for each $s \in S$. The main theorem of [3] shows that X can be covered by a collection of vertex disjoint paths. A nice implication of their result is that Theorem 1 holds even if S and T were not contained in levels, but were antichains.

We also note that the routing perspective in §4 answers a question of Sachdeva from a collection of open problems on Boolean functions [27] (Pg 19, "Routing on the hypercube"). We discuss more in §4.

2 Alternate proof of the Lehman-Ron Theorem

As a warm-up, we set up the main idea with a proof of the Lehman-Ron theorem. We begin with an important definition.

▶ **Definition 4.** *Given two sets S and T of the directed hypercube, the* cover *graph $G_{S,T}$ is formed by the union of all paths from S to T.*

In other words, the cover graph is the subset of the hypercube, that contains all vertices v such that $s \prec v \prec t$ (for $s \in S, t \in T$). The cover graph inherits "layers" via intersection with the original hypercube layers. In particular, layer L_i of the cover graph is only S and layer L_j is only T.

For the sake of contradiction, consider the *minimal counterexample* of Theorem 1 in terms of $|S| + |T|$. Consider the following flow network which contains $V(H)$ and also supernodes Ⓢ and Ⓣ. Ⓢ has a directed edge to every vertex in $s \in S$, and every vertex $t \in T$ has a directed edge to Ⓣ. We construct a flow network by setting the following *vertex* capacities to $G_{S,T}$: the supernodes have infinite capacity while every vertex in $V(H)$ has capacity 1. Since (S, T) is a counterexample, by the theory of flows and cuts, the maximum Ⓢ,Ⓣ flow in this vertex-capacitated network is $< |S|$. And so, using flow-cut duality, we know that there exists a *cut* $C \subseteq V(H)$ such that (a) $|C| < |S|$, (b) *every* path from Ⓢ to Ⓣ contains a C-vertex. Call a path *cut-free* if it doesn't contain a vertex from C. We can partition all vertices into three sets $\mathcal{S}, C, \mathcal{T}$, where $\mathcal{S}$ contains all vertices that are reached by a cut-free path from Ⓢ, and vertices of $\mathcal{T}$ can reach Ⓣ by a cut-free path. In particular, there is no edge from a vertex in $\mathcal{S}$ to a vertex in $\mathcal{T}$; all edges leaving $\mathcal{S}$ enter C, and all edges entering $\mathcal{T}$ originate from C. We make a quick observation using the minimality of our counterexample.

▶ **Lemma 5.** *C is disjoint from $S \cup T$.*

Proof. If C contains a vertex $S \cup T$, then one obtains a smaller counterexample. If $v \in C \cap S$, then $S' := S \setminus \{v\}$, $T' := T \setminus \{\phi(v)\}$ and $\phi' := \phi_{|S'}$ forms a matched pair $(S', T'; \phi')$ which is also a counterexample: the cut $(C - \{v\})$ is a valid cut of value $|C| < |S| - 1 = |S'|$. So, $C \cap S = \emptyset$. The proof of $C \cap T = \emptyset$ is analogous. ◀

Our setup so far is a restructuring of the original Lehman-Ron proof. The following lemma is where we start to differ. This lemma is a consequence of *complementary slackness* from the theory of linear optimization, and is the central tool for our new proof.

▶ **Lemma 6.** *There exists a collection of* vertex disjoint *paths $\mathcal{P}$ where every path $p \in \mathcal{P}$ begins at a vertex in S and ends at a vertex in T and*

- *Every path $p \in \mathcal{P}$ contains* exactly *one vertex in C.*
- *Every vertex $v \in C$ is in* exactly *one path in $\mathcal{P}$.*

Note that all these paths are in the cover graph $G_{S,T}$. Using the above collection of paths, we make a key definition.

▶ **Definition 7.** *A vertex v is a* gateway *if (i) $v \in \mathcal{S}$, (ii) v doesn't lie on any path in $\mathcal{P}$, and (iii) there is at least one edge (v, w) in the cover-graph.*

The Lehman-Ron theorem, Theorem 1, follows directly from the following lemma.

▶ **Lemma 8.** *For all $i \le k \le j-1$, the kth layer L_k of the cover-graph contains a gateway vertex.*

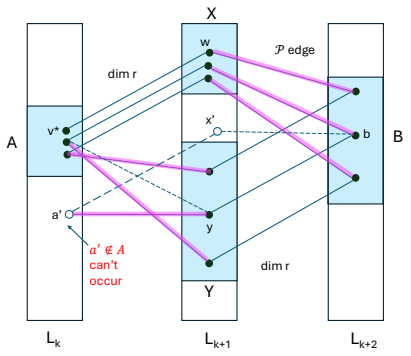
Proof of Theorem 1. Consider the gateway vertex $v \in L_{j-1} \cap \mathcal{S}$ with edge (v, w) in the cover-graph. Note $w \in T$ and therefore in $\mathcal{T}$. There is an edge from $\mathcal{S}$ to $\mathcal{T}$. Contradiction. Hence, there is no (minimal) counterexample to Theorem 1. ◀

Before giving the formal proof of Lemma 8 directly, let us describe the main idea which uses the symmetry of the hypercube. First, let us observe the layer L_i, that is S, contains a gateway vertex s. Indeed, there are $\le |S| - 1$ paths in $\mathcal{P}$ and so there is some $s \in S$ not in any of these paths. Furthermore, all edges that lead s to $\phi(s)$ lie in the cover-graph.

Now, let's see how to get a gateway in the next layer L_{i+1}. Consider any edge $(s, x^{(1)})$ in $G_{S,T}$, and suppose this edge corresponds to projecting according to some dimension r. That is $s_r = 0$ and $x^{(1)}_r = 1$. If $x^{(1)}$ is not in any path in $\mathcal{P}$, we have discovered the desired gateway in L_{i+1}. Otherwise, $x^{(1)}$ lies on some path, say, $P \in \mathcal{P}$. Follow P *forwards* for a single edge from $x^{(1)}$, to get to $x^{(2)}$. Note that $x^{(2)} \in L_{i+2}$ and has rth-coordinate 1. Now, one can project "down" on the r-coordinate to get $x^{(3)} \in L_{i+1}$. Observe that $(s, x^{(3)})$ is a *projection* of the edge $(x^{(1)}, x^{(2)})$ along the rth-dimension and hence is an edge of the cover-graph. Next, observe that $x^{(3)}$ cannot be in $\mathcal{T}$, so either $x^{(3)} \in \mathcal{S}$ or $x^{(3)} \in C$. If $x_3 \in \mathcal{S}$ and not on any path in $\mathcal{P}$, we are done. Otherwise $x^{(3)}$ lies on some other path $Q \in \mathcal{P}$. We now walk *backwards* along Q, to get $x^{(4)} \in S$. Noting that $x^{(4)}$ has r-coordinate 0, we can redo the entire process above. Observe that each "step" proceeds along a *matching*. Either we project, walk from L_{i+1} to L_{i+2} using a path in $\mathcal{P}$, or walk backward from L_{i+1} to L_i using a path in $\mathcal{P}$. Each of these is using a matching edge, and no vertex is ever visited twice in the entire process. Hence, this process must terminate, at which point a gateway is discovered. And then one uses the same idea to obtain a gateway vertex in L_{i+2}, and so on. One can convert this idea into a formal proof, but it becomes notationally cumbersome. A cleaner proof method is to consider a potential "fixed point" of this process, and prove a contradiction.

Proof of Lemma 8. Fix a collection of paths $\mathcal{P}$ as given by Lemma 6. As argued above, L_i has a gateway vertex. Let $k \in [i, j-1]$ be the largest value such that L_k contains a gateway vertex. If $k = j-1$, we are done. So suppose that $k < j-1$. We now engineer a contradiction. Let $v^\star$ be a gateway in L_k. So $v^\star \in \mathcal{S}$ and has an edge $(v^\star, w)$ leaving it. Let r be the dimension of this edge implying $v^\star_r = 0$ and $w_r = 1$. Let Π_r be the projection operator which flips the rth coordinate; so $\Pi_r(v^\star) = w$ and vice versa. Define the following sets; we give a illustration for convenience where the pink highlighted edges participate in paths of $\mathcal{P}$.

- $A = \{a \ : \ a \in \mathcal{S} \cap L_k, a_r = 0 \text{ and } \Pi_r(a) \in G_{\mathcal{S},\mathcal{T}}\}$.
- $X = \Pi_r(A) = \{\Pi_r(a) \ : \ a \in A\}$.
- $B = \{b \ : \ (x,b) \in P, \text{ for some } x \in X, P \in \mathcal{P}\}$.
- $Y = \Pi_r(B) = \{\Pi_r(b) \ : \ b \in B\}$.



Observe that $X \cup Y \subseteq L_{k+1}$, and $B \subseteq L_{k+2}$. We next argue these subsets lie in the cover-graph. By definition of A, $X \subseteq G_{\mathcal{S},\mathcal{T}}$, and since B is obtained via paths from X, we get $B \subseteq G_{\mathcal{S},\mathcal{T}}$. Finally, for any $y \in Y$, note that $b = \Pi_r(y)$ lies in B, and there's a path (a, x, b) for some $a \in A$ and $x = \Pi_r(a)$. But note that (a, y) edge is the r-projection of (x, b), and so (a, y, b) is a path implying $y \in G_{\mathcal{S},\mathcal{T}}$. Thus, Y also lies in the cover-graph.

Since X is a projection of A, we get $|X| = |A|$. Consider any $x \in X$. Since $x \in L_{k+1}$, by our choice of k, x isn't a gateway vertex. Since (a, x) is an edge for $a = \Pi_r(x)$, we have $x \in \mathcal{S}$ or $x \in C$. In the former case since x isn't a gateway vertex, and in the latter case by Lemma 6, there exists $P \in \mathcal{P}$ with $x \in P$. And so, $|B| = |X|$. By projection property, $|Y| = |B|$, and so following the above chain of equalities, we get $|Y| = |A|$.

Next consider any $y \in Y$. As argued above, there is a vertex $a \in A$ such that (a, y) is an edge in the cover-graph (the projection of $(x, b) \in P$ where $b = \Pi_r(y)$). Since $y \in L_{k+1}$, by our choice of k, y isn't a gateway vertex. Since (a, y) is an edge, we must have $y \in \mathcal{S}$ or $y \in C$. In the former case since y isn't a gateway vertex, and in the latter case by Lemma 6, there exists $Q \in \mathcal{P}$ with $y \in Q$. Let (a', y) be the edge in Q taking y "backward" along Q. We claim that $a' \in A$. If $y \in \mathcal{S}$, then $a' \in \mathcal{S}$ since (a', y) is an edge; if $y \in C$ then by Lemma 6, $a' \in \mathcal{S}$ (the path Q doesn't contain two cut-vertices). Since $(a', x' = \Pi_r(a'))$ lies in the cover graph since the r-projection of the (a', y) lies there, we get x' lies in the cover-graph, implying a' must lie in A. The above figure illustrates this. In short, there are $|Y|$ paths of $\mathcal{P}$ that contain vertices of A. Since $|Y| = |A|$ and since these paths are all vertex-disjoint, we conclude *all* vertices in A are present in some path of $\mathcal{P}$. However, the gateway vertex $v^\star \in A$ and doesn't lie on any path of $\mathcal{P}$. Contradiction. ◀

3 Proof of the generalization Theorem 2

We now prove the generalization of the Lehman-Ron theorem using the proof strategy above. As before, we start with a minimal counter-example and use it to construct a flow network. We have the same directed graph as in the previous section with supernodes Ⓢ and Ⓣ, with every edge incident to supernodes having infinite capacity. The crucial difference is that we have both vertex and edge capacities. Each edge has capacity one, and each vertex has capacity two. (Alternately, it costs "one unit" to cut an edge, but "two units" to cut a vertex.)

Since we have a counter-example, again using the theory of flows, the maximum flow in this network is $< 2|S|$. And thus, by duality, we posit that there exists a *pair* (C, F) with $C \subseteq V(H)$ and $F \subseteq E(H)$, such that (a) $2|C| + |F| < 2|S|$, (b) every path from s to t either contains a C-vertex or an F-edge or both. A path from a vertex u to v is now called *cut-free* if it contains neither a vertex from C nor an edge from E. As before, we can partition all

vertices into three sets $\mathcal{S}, C, \mathcal{T}$, where $\mathcal{S}$ contains all vertices that are reached by a cut-free path from S, and vertices of $\mathcal{T}$ can reach T by a cut-free path. Note that the edges of F are from vertices in $\mathcal{S}$ to vertices in $\mathcal{T}$. And, as before, by minimality of the counter-example, the following simple observation holds.

▶ **Lemma 9.** *(i) The cut set C is disjoint from $S \cup T$. (ii) There exists a mincut (C, F) such that no vertex participates in more than one edge of F.*

Proof. Proof of (i) is exactly as in Lemma 5. Suppose v participates in at least two edges of F. Observe that any S-T path through any of these edges must go via v. Hence, we can remove these edges from F, add v to C, and preserve the fact that $C \cup F$ is an S-T cut. Moreover, the cut value does not increase. ◀

We can now give the analog of Lemma 6.

▶ **Lemma 10.** *There exists a collection of paths $\mathcal{P}$ with the following properties.*

- *Every path $p \in \mathcal{P}$ begins at a vertex in S and ends at a vertex in T.*
- *The paths are pairwise edge-disjoint and any vertex is in at most two paths.*
- *Every path $p \in \mathcal{P}$ either contains* exactly *one vertex in C or* exactly *one edge in F, but not both.*
- *Every vertex $v \in C$ is in* exactly *two paths in $\mathcal{P}$ and every edge $e \in F$ is in* exactly *one path of $\mathcal{P}$.*

Proof. By complementary slackness (Theeorem A.7 of [23]) , every maximum flow must saturate the min cut. Together with integrality of flow, this implies the existence of an integral flow saturating $C \cup F$. We give a (simple) formal explanation. By integrality of flow, there is a maximum flow that can be decomposed into paths. Let $\mathcal{P}$ be those paths. Since these paths form a feasible flow, they satisfy the first two bullet points of the lemma. By duality, $|\mathcal{P}| = 2|C| + |F|$. For each path $p \in P$, let c_p be the number of cut elements in $C \cup F$ that the path contains. For each cut element $e \in C \cup F$, let k_e be the number of paths that e participates in. So $\sum_{p \in \mathcal{P}} c_p = \sum_{e \in C \cup F} k_e$. Note that $\forall p, c_p \geq 1$, since $C \cup F$ is a valid cut. Thus, $\sum_{p \in \mathcal{P}} c_p \geq |\mathcal{P}|$. Now, observe that $\forall e \in C, k_e \leq 2$ and $\forall e \in F$, $k_e \leq 1$, since $C \cup F$ must satisfy the flow constraints. Hence, $\sum_{e \in C \cup F} k_e \leq 2|C| + F$. We get $|\mathcal{P}| \leq \sum_{p \in \mathcal{P}} c_p = \sum_{e \in C \cup F} k_e = 2|C| + |F| = |\mathcal{P}|$. Thus, the inequalities above are all equalities. So $\forall p, c_p = 1$ (third bullet) and $\forall e \in C, k_e = 2$ and $\forall e \in F$, $k_e = 1$ (fourth bullet). ◀

Next we provide the relevant generalization of gateway vertices earlier defined in Definition 7

▶ **Definition 11.** *A vertex v is a* gateway *if (i) $v \in \mathcal{S}$, (ii) v lies on at most one path in $\mathcal{P}$, and (iii) there is at least one edge $(v, w) \notin F$ leaving v in $G_{\mathcal{S},\mathcal{T}}$.*

As before, the proof of Theorem 2 follows from the following lemma, and the remainder of this section will prove it.

▶ **Lemma 12.** *For all $i \leq k \leq j - 1$, L_k contains a gateway vertex.*

As in the proof of Lemma 8, we proceed via minimal counterexamples. First, we establish that there is a vertex in L_i that is a gateway vertex. There are $< 2|S|$ paths in $\mathcal{P}$. Since any vertex participates in at most 2 paths, some vertex in s participates in at most 1 path. Since $\phi(s)$ is at least distance 2 away from s, s has degree at least 2 in $G_{S,T}$. (This is where the distance between S and T is used.) At most one of those edges is in F (Lemma 9), so there is some edge leaving s that is not in F. Thus, s is a gateway vertex.

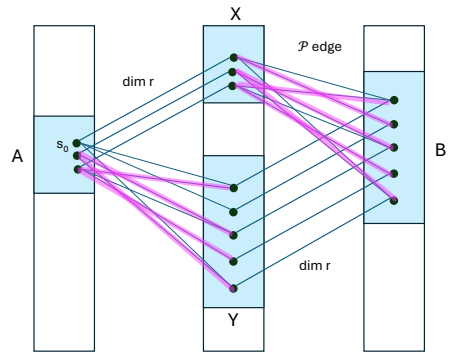


Figure 1 The setup of the proof of Lemma 12. The set X is a projection of A along dimension r, and similarly Y is a (downward) projection of B. The edges between A and Y are projections of edges between X and B.

Now, let $k \in [i, j-1]$ be the largest value such that L_k contains a gateway vertex. If $k = j-1$, we are done. So suppose that $k < j-1$, and we will engineer a contradiction. Let $v^\star$ be a gateway vertex in L_k. So $v^\star \in \mathcal{S}$, participates in at most one path of $\mathcal{P}$, and there is some edge leaving $v^\star$ that is not in F. First, let us choose the projection dimension r as follows. If $v^\star$ lies on exactly one path $P \in \mathcal{P}$, then let $(v^\star, w)$ be the edge of P incident to $v^\star$ and let r be the coordinate that this edge flips. If $v^\star$ lies on no path $P \in \mathcal{P}$, then since $k < j-1$, there are at least two[2] edges incident to $v^\star$ that are in the cover-graph; let $(v^\star, w)$ be any such edge and let r be the coordinate of this edge. As in the proof of Lemma 8, we let Π_r denote the projection operator, and we define the sets exactly as in the previous proof.

- $A = \{a \ : \ a \in \mathcal{S} \cap L_k, a_r = 0 \text{ and } \Pi_r(a) \in G_{\mathcal{S},\mathcal{T}}\}$. Note that $v^\star \in A$.
- $X = \Pi_r(A) = \{\Pi_r(a) \ : \ a \in A\}$.
- $B = \{b \ : \ (x, b) \in P, \text{ for some } x \in X, P \in \mathcal{P}\}$.
- $Y = \Pi_r(B) = \{\Pi_r(b) \ : \ b \in B\}$.

As in the proof of Lemma 8, we have $X \cup Y \subseteq L_{k+1}$, and $B \subseteq L_{k+2}$ and that A, X, B, Y all lie in the cover graph $G_{\mathcal{S},\mathcal{T}}$. As there, we crucially use the property that the projection of every (A, Y) edge lies is an (X, B) edge. Note that, we have $|A| = |X|$ and $|B| = |Y|$. However, unlike in Lemma 8, there can be *two* paths of $\mathcal{P}$ that may contain $x \in X$, and thus, $|B|$ may be larger in size than $|X|$. And this makes the proof slightly more complicated.

We now restrict our attention to these four sets. As a result, we only consider the cover graph $G_{A,B}$, which, recall, is the union of all paths from vertices in A to vertices in B (irrespective of whether they contain cut-elements). Both A and B lie in this graph by definition. We assert $X \subseteq G_{A,B}$. Take a vertex $x \in X$. If $x \in P$ for some path $P \in \mathcal{P}$ then it lies in $G_{A,B}$ by construction. We now show if x is not any path in $\mathcal{P}$, we already have a contradiction. Indeed, the edge (a, x) where $a = \Pi_r(x)$ isn't in F for otherwise, by Lemma 10 there would be a path containing (a, x). Similarly $x \notin C$. So, $x \in \mathcal{S}$. Since $x \in L_{k+1}$ and $k < j-1$, there is at least one edge leaving it, and as before, this edge isn't in F. So, x is a gateway vertex which contradicts the choice of k. Thus, the subset X lies in $G_{A,B}$. And since every (A, Y) edge is an r-projection of an (X, B) edge, we argue that Y also lies in $G_{A,B}$.

We now make another definition which will lead to our contradiction.

[2] There is some $s \in S$ such that $s \prec v \prec \phi(s)$ and there are at least 2 edge disjoint paths from v to $\phi(s)$.

▶ **Definition 13.** *An edge in $G_{A,B}$ is called* pink *if it lies in $\mathcal{P}$ and does not change the r coordinate. For any subset of vertices W in $G_{A,B}$, pink(W) is the number of pink edges W is incident to. For singleton subsets $\{v\}$ we abuse notation and call pink$(\{v\})$ simply pink(v).*

Thus, every pink edge incident to A is incident to Y, and every pink edge incident to X is incident to B. The following claim is a consequence.

▷ Claim 14. $\text{pink}(A) = \text{pink}(Y)$ and $\text{pink}(X) = \text{pink}(B)$.

Next, observe that the gateway $v^\star$ participates in at most one path of $\mathcal{P}$. If such a path existed, we chose the dimension r according to the incident edge of this path. Hence, $\text{pink}(v^\star) = 0$. This is central to proving the final contradiction.

▷ Claim 15. $\text{pink}(X) > \text{pink}(A)$.

Proof. There is a perfect matching between A and X. We prove that $\forall a \in A$, $\text{pink}(\Pi_r(a)) \geq \text{pink}(a)$. Furthermore, for the gateway $v^\star$, we get a strict inequality $\text{pink}(\Pi_r(v^\star)) > \text{pink}(v^\star)$. It is convenient to do a case analysis based on whether $x = \Pi_r(a)$ is in $\mathcal{S}, C$, or $\mathcal{T}$. Note that $\text{pink}(a) \leq 2$ by Lemma 10 since any vertex participates in at most 2 paths of $\mathcal{P}$. Note that all edges of $\mathcal{P}$ that leave X lead to B (by construction), so all these edges are in $G_{A,B}$.

$x \in C$: This is the easiest. By Lemma 10, there are two paths through x. So $\text{pink}(x) = 2 \geq \text{pink}(a)$.

$x \in \mathcal{T}$: Note that $a \in \mathcal{S}$, and so in this case, the r-projection edge (a, x) must be $\in F$. By Lemma 10, there is a path of $\mathcal{P}$ through (a, x), and the next edge leaving x goes into B. Since the r-coordinate doesn't change, this edge is pink, and so $\text{pink}(x) \geq 1$. Since a participates in the path through projection edge (a, x), which is *not* pink, it can participate in at most one other path of $\mathcal{P}$ (by Lemma 10, no vertex is in more than 2 paths). Thus, in this case $\text{pink}(a) \leq 1$ implying $\text{pink}(a) \leq \text{pink}(x)$.

$x \in \mathcal{S}$: By our assumption, x cannot be a gateway vertex. So either all edges leaving x are in F or x is in 2 paths of $\mathcal{P}$. If the latter happens, then $\text{pink}(x) = 2$, completing this case. So let us assume the former. There must be some edge leaving x in $G_{A,B}$ because x is in the cover graph $G_{A,B}$; since x is not a gateway vertex, this edge which must be in F. By Lemma 10, there is a path $P \in \mathcal{P}$ containing this and the edge incident to x doesn't change the r-coordinate, and thus is pink. So, $\text{pink}(x) \geq 1$. So, if $\text{pink}(a) \leq 1$ we are done. So, suppose $\text{pink}(a) = 2$. That is, there are two distinct edges (a, y) and (a, y') which are pink. Consider the r-projection of these edges, $(x, \Pi_r(y))$ and $(x, \Pi_r(y'))$; these are present in $G_{A,B}$ and so by our assumption above, these two must be in F. But again by Lemma 10, there are two paths in $\mathcal{P}$ containing these, and so these edges are pink, implying $\text{pink}(x) = 2$ as well. This settles this case.

All in all, we have proven that $\text{pink}(x) \geq \text{pink}(a)$, and note that in all cases, $\text{pink}(x) \geq 1$. Since $\text{pink}(v^\star) = 0$, we get the strict inequality $\text{pink}(\Pi_r(v^\star)) > \text{pink}(v^\star)$, completing the proof of the claim. ◁

The next claim which, along with Claim 14, contradicts Claim 15, completing our proof of Lemma 12.

▷ Claim 16. $\text{pink}(B) \leq \text{pink}(Y)$.

Proof. There is a perfect matching between B and Y by the Π_r projection. We will show that for every $b \in B$, $\text{pink}(b) \leq \text{pink}(y)$, where $y = \Pi_r(b)$. The proof is analogous to that of Claim 15, with a subtle difference. All edges of $\mathcal{P}$ leaving X are in $G_{A,B}$ by construction. But all edges of $\mathcal{P}$ entering Y might not be in $G_{A,B}$. In particular, there could be a path $Q \in \mathcal{P}$ which contains y and the predecessor, call it z, of y on this path may not be in A. For instance, this could occur if $y \in \mathcal{T}$ and $z \in C$ (and thus not in A).

With hindsight, we assert that if $y \in \mathcal{S} \cup C$, then the vertex z indeed lies in A. To see this note that $z \in \mathcal{S}$; this follows from Lemma 10 since the path Q contains exactly one cut-vertex or cut-edge. Furthermore, $\Pi_r(z)$ must lie in $G_{\mathcal{S},\mathcal{T}}$ since $(z, \Pi_r(z), b)$ is present in the hypercube. In all, $z \in A$. The reader may wonder why this subtlety didn't arise in the original Lehman-Ron proof that we showed in the previous section. Well, there is a vertex $a \in A$ such that (a, y) is an edge, and since in the previous section we only had vertex cuts, we could (and did) assert $y \in S \cup C$. But now the edge (a, y) could be in F. As we will see, the case when $y \in T$ is actually easy to take care of.

Now for the case analysis. Recall that $b \in B$ and $y = \Pi_r(w) \in Y$.

$y \in C$: By Lemma 10 there are two paths entering y, and since $y \in C$ due to the discussion about the subtlety above, $\text{pink}(y) = 2$. Note that $\text{pink}(b) \leq 2$ since there can be at most two paths of $\mathcal{P}$ incident on b.

$y \in \mathcal{S}$: Since y is not a gateway vertex (overall assumption), either y participates in two paths of $\mathcal{P}$ or all edges leaving y are in F. In the former case, since $y \in S$ due to the discussion about the subtlety above, $\text{pink}(y) = 2$. In the latter case, the edge (y, b) must be in F. So there is at least one path through y and, again since $y \in S$, $\text{pink}(y) \geq 1$. Now note that the (y, b) edge is *not* pink although it is on a path in $\mathcal{P}$ because the r-coordinate changes. So, $\text{pink}(b) \leq 1$.

$y \in \mathcal{T}$: Observe that all edges (z, y) with $z \in A \subseteq \mathcal{S}$ must be cut, that is $(z, y) \in F$. By Lemma 9, all such edges are on paths in $\mathcal{P}$ and are thus pink (they are clearly in $G_{A,B}$ and don't change r-coordinate). Suppose $\text{pink}(b) = t$ where $t \in \{1, 2\}$. Then there are t distinct pink edges of the form (x, b) with these x's in X. Consider their r-projections, that is, the t edges of the form $(\Pi_r(x), y)$. Since $\Pi_r(x) \in A$, all these edges must be pink. This shows that if $y \in \mathcal{T}$, $\text{pink}(y) \geq \text{pink}(b)$. ◁

4 Connections to monotonicity testing

The motivation for Conjecture 3 (and indeed, Theorem 1) is a deeper understanding of the problem of monotonicity testing of functions, a problem which, especially over the hypercube and hypergrid domains, has had a rich history of more than 25 years [38, 25, 31, 24, 36, 29, 32, 1, 33, 2, 28, 40, 6, 16, 26, 13, 39, 7, 18, 19, 21, 5, 14, 20, 17, 35, 4, 22, 8, 9, 12, 34, 15, 11, 10]. A function $f : \{0,1\}^d \to \{0,1\}$ is monotone if $\forall x, y \in \{0,1\}^d$ where $x \prec y$, $f(x) \leq f(y)$. The distance between two functions f, g is $|\{x : f(x) \neq g(x)\}|/2^d$, and the distance of f to monotonicity, denote ε_f, is the minimum distance of f to a monotone g. A function f is said to be ε-far from monotone if $\varepsilon_f \geq \varepsilon$. The aim of a tester is to distinguish a monotone function from one that is "far" from monotone. There is a special focus on non-adaptive monotonicity testers with one-sided error. These are testers that (i) always accept monotone functions, and (ii) make all their queries in advance. After a long line of work, this has been resolved (up to $\log d, \text{poly}(\varepsilon^{-1})$ factors) to be $\Theta(\sqrt{d})$ [31, 29, 19, 21, 35, 20, 22].

The study of these monotonicity testers led to the discovery of *directed isoperimetric inequalities*. Much of the study in this paper came from attempts at an alternate, more combinatorial proofs of a central isoperimetric inequality, the so-called *robust directed Talagrand theorem* due to Khot, Minzer, and Safra [35]. In this section, we give connections between monotonicity testing, directed isoperimetric inequalities (like the KMS theorem), and routing on the directed hypercube. Most importantly, we describe another routing statement, Conjecture 26, which implies the KMS theorem. We believe that Conjecture 26 and Conjecture 3 are closely related, as explained in §4.2.

▶ **Definition 17.** *A pair of vertices of* $\{0,1\}^d$ $x \prec y$ *is called a* violation *of* $f(x) > f(y)$. *This pair is called a* violated edge *if additionally,* (x, y) *is an edge of the hypercube. For any* x*, the directed influence* $\mathrm{Inf}_f^+(x)$ *is the number of violated edges incident to* x*. The directed influence of* f*, denoted* I_f^+*, is* $2^{-d}\sum_x \mathrm{Inf}_f^+(x)$.

The most basic inequality is the directed Poincare inequality, which directly leads to $O(d)$ query monotonicity testers (for constant ε).

▶ **Theorem 18** ([31])**.** *For any* $f : \{0,1\}^d \to \{0,1\}$, $I_f^+ \geq \varepsilon_f$.

The main step towards $o(d)$ query testers (for constant ε) is a stronger isoperimetric inequality, the directed Margulis bound. Let Γ_f^+ denote the size of the largest matching of violated edges, which is a measure of the vertex boundary.

▶ **Theorem 19** ([19])**.** *For any* $f : \{0,1\}^d \to \{0,1\}$, $I_f^+ \cdot \Gamma_f^+ = \Omega(\varepsilon_f^2)$.

The culmination of this line of work lead to the *robust, directed Talagrand inequality* for KMS, which yielded the (near) optimal $\widetilde{O}(\sqrt{d})$-query non-adaptive monotonicity tester. (The original KMS result lost a log factor, which was removed by Pallavoor-Raskhodnikova-Waingarten [37].)

▶ **Theorem 20** ([35, 37])**.** *Let* χ *be any bicoloring of the directed hypercube edges, with two colors* 0 *and* 1*. For any* $f : \{0,1\}^d \to \{0,1\}$ *and for any* $x \in \{0,1\}^d$*, let* $\mathrm{Inf}_{f,\chi}^+(x)$ *be the number of violated edges incident to* x *whose color is* $f(x)$*. Then,*

$$2^{-d}\sum_x \sqrt{\mathrm{Inf}_{f,\chi}^+(x)} = \Omega(\varepsilon_f)$$

In the next subsection, we give combinatorial interpretations to each of these statements. The reason for Conjecture 3 and a deeper study of hypercube routing was to get alternate proofs of Theorem 20. A big mystery of all these directed isoperimetric inequalities is the appearance of ε_f, the distance to monotonicity, as a "directed version" of the variance of f. It appears as if ε_f is the "right measure" of directed volume. We hope that alternate proofs of Theorem 20 may shed some light on this mystery.

4.1 From flows to directed isoperimetry

In what follows, all flow networks are over the directed hypercube. There is a source set S, and the aim is to route flow to the complement $\overline{S}$[3]. In the various routing theorems, we set different edge/vertex capacities and try to lower bound the maximum flow from S to $\overline{S}$. In all the flow settings, we have unit edge capacities.
We start with a notion of the "directed volume" of a set.

▶ **Definition 21.** *For* $S \subseteq \{0,1\}^d$*, the directed volume of* S*, denoted* $\mu^+(S)$ *is*

$$\max_{\substack{S' \subseteq S \\ T' \subseteq \overline{S}}} \left\{ |S'| \;\middle|\; \exists \phi, (S', T'; \phi) \text{ is a matched pair} \right\}$$

Any matched pair $(S', T'; \phi)$ *that attains the maximum is called a* directed volume certificate.

[3] Formally, one creates a supernode Ⓢ that connects to S, and a supernode Ⓣ with connections from T. All these connections have infinite capacity.

We now explain why the directed Poincare inequality of Theorem 18 essentially shows that one can send $\mu^+(S)$ units of flow from S to $\overline{S}$ with unit edge capacities. This is a simple application of the max-flow-min-cut theorem, and we provide the proof for completeness.

▶ **Theorem 22.** *Consider the directed hypercube flow network with unit edge capacities, and source set S. The maxflow is at least $\mu^+(S)$.*

Proof. Consider the indicator Boolean function $f : \{0,1\}^d \to \{0,1\}$ where $f(x) = 1$ iff $x \in S$. Using a standard connection between distance to monotonicity (Corollary 2 of [29]) one can argue that $\varepsilon_f = \mu^+(S)/2^d$. Any Ⓢ-Ⓣ cut must remove all edges from S to $\overline{S}$. These are precisely the violated edges of f, which are at least $\varepsilon_f 2^d = \mu^+(S)$ many (Theorem 18). The theorem follows from the duality between max-flow and min-cut. ◀

Thus, the basic directed Poincare inequality basically gives a flow bound for the directed hypercube flow network. We will now interpret the more sophisticated isoperimetric theorems as more general flow statements. A crucial notion is the separation distance of a set.

▶ **Definition 23.** *For any matched pair $(S, T; \phi)$, the separation distance is $|S|^{-1} \sum_{s \in S} (|\phi(s)| - |s|) = |S|^{-1} \Big(\sum_{t \in T} |t| - \sum_{s \in S} |s| \Big)$. Here $|x|$ denotes the number of 1s in $x \in \{0,1\}^d$. The* **separation distance** *of S is the smallest separation distance over directed volume certificates of S.*

A key theorem of [19] shows that larger separation distance implies more (edge disjoint) flow. This theorem is a strengthening of the directed Poincare inequality of Theorem 18, and essentially a flow rewording of Lemma 2.6 of [19]. The proof is entirely analogous to that of Theorem 22 and is omitted.

▶ **Theorem 24** (Lemma 2.6, [19])**.** *Consider the directed hypercube flow network with unit edge capacities, with source set S having separation distance r. The maxflow is at least $r\mu^+(S)$.*

What if we desire *vertex* disjoint paths? Lemma 2.5 of [19] answers this question, and the central tool is the Lehman-Ron theorem. Together, the two theorems above directly imply the directed Margulis statement of Theorem 19.

▶ **Theorem 25** (Lemma 2.5, [19])**.** *Consider a flow network with unit vertex capacities, with source set S having separation distance r. The maxflow is at least $\mu^+(S)/32r$.*

This brings us to a sort of "intellectual starting point" for this paper. The theorems above clearly show how directed isoperimetry and flows are intimately connected. Moreover, statements like Theorem 19 suggest relations between flows with edge capacities, and flow with vertex capacities. We were motivated to see if the KMS theorem (Theorem 20) could be proven from a flow perspective.

▶ **Conjecture 26.** *Let source set S have separation distance r. Consider a flow network with unit edge capacities and vertex capacities r^2. The maxflow is at least $\Omega(r\mu^+(S))$.*

Note that the above is a *simultaneous* strengthening of Theorem 24 and Theorem 25; if we remove either the edge capacity restriction or the vertex capacity restriction, then we get the above theorems. We show that Conjecture 26 implies the robust Talagrand isoperimetry theorem.

▷ Claim 27. Conjecture 26 implies Theorem 20.

Proof. Consider a Boolean function $f : \{0,1\}^d \to \{0,1\}$. Consider any bicoloring χ of the violated edges. Our aim is to lower bound $\sum_x \sqrt{\mathrm{Inf}^+_{f,\chi}(x)}$.

Let S be the set of 1-valued points. By Conjecture 26 and the maxflow-mincut theorem, the mincut of the flow network (where edges have capacity 1 and vertices have capacity r^2) is at least $Cr\mu^+(S)$ for some constant $C > 0$. Note that all edges from S to $\overline{S}$ must be cut; moreover any separation of (the endpoints of) these edges is a valid $(S, \overline{S})$ cut. In terms of f, these are precisely the violated edges.

Let us use χ to construct a cut. For convenience, let $d(x)$ denote $\mathrm{Inf}^+_{f,\chi}(x)$. If $d(x) \le r^2$, we cut all violated edges incident to x. Otherwise, we cut the vertex x. The total cut value is $\sum_{x:d(x)\le r^2} d(x) + r^2|\{x \mid d(x) > r^2\}|$. By Conjecture 26, the cut value is at least $Cr\mu^+(S)$. We split into two cases.

Case 1, $\sum_{x:d(x)\le r^2} d(x) \ge Cr\mu^+(S)/2$. Observe that $\sum_{x:d(x)\le r^2} \sqrt{d(x)\cdot d(x)} \le r\sum_{x:d(x)\le r^2} \sqrt{d(x)}$. Thus, $\sum_x \sqrt{\mathrm{Inf}^+_{f,\chi}(x)} \ge C\mu^+(S)/2 = C\varepsilon_f 2^d/2$.

Case 2, $\sum_{x:d(x)\le r^2} d(x) < Cr\mu^+(S)/2$. So $r^2|\{x \mid d(x) > r^2\}| \ge Cr\mu^+(S)/2$, implying $|\{x \mid d(x) > r^2\}| \ge C\mu^+(S)/(2r)$. We can lower bound $\sum_x \sqrt{d(x)} \ge r\sum_{x:d(x)>r^2} \ge C\mu^+(S)/2 = C\varepsilon_f 2^d/2$. ◁

We believe that Conjecture 26 is stronger than Theorem 20, because it explicitly involves the separation distance of S.

As an aside, the connection between flows and directed isoperimetry resolves an open question in [27] (Pg 19, "Routing on the hypercube"). It is actually a direct consequence of Theorem 18.

▶ **Theorem 28.** *Let (S, T, ϕ) be a matched pair where S and T are disjoint. There exist $|S|$ monotone edge disjoint paths from S to T.*

Proof. Consider the directed hypercube and take a mincut separating S from T. Construct a Boolean function that assigns 1 to the "S-side", and 0 to the "T-side". All remaining vertices can be assigned values such that they do not participate in any monotonicity violation. Note that these vertices cannot be on a directed path from S to T. (Process vertices according to the partial order. For x, set $f(x)$ to be $\max_{y\prec x: f(y) \text{ assigned}} f(y)$. If no $f(y)$ is assigned, set $f(x) = 0$. Observe that if $f(x)$ is assigned value 1, then x must be greater than some point in S. This means that x cannot be less than any point in T, and hence does not create monotonicity violations.)

This function has distance to monotonicity at least $|S|/2^n$. So by Theorem 18, there are at least $|S|$ edges which have value $(1, 0)$. These are precisely cut edges, from the S-side to the T-side. Hence, the cut value is at least $|S|$. Set up a flow problem on the directed hypercube where every edge has unit capacity, vertices in S are sources, and vertices in T are sinks. By the maxflow-mincut theorem, there is a flow of value at least $|S|$. This flow gives edge-disjoint paths from S to T. ◀

4.2 Connections between conjectures

From the perspective of monotonicity testing and directed isoperimetry, Conjecture 26 is more important. From a purely combinatorial (and maybe aesthetic) viewpoint, Conjecture 3 is more appealing. We believe that a proof of Conjecture 3 will shed light on Conjecture 26. This section is speculative, but gives some of the original motivations for studying Conjecture 3.

An uncrossing argument of [19] relates general matched pairs to matched pairs contained in level sets. (These arguments are in Section 2.4 of [19], especially Claim 2.7.2 and Claim 2.7.3.)

▶ **Lemma 29.** *Consider a set S with separation distance r. There exist a collection of matched pairs $(S_1, T_1, \phi_1), (S_2, T_2, \phi_2), \ldots$ with the following properties.*

- $\bigcup_i S_i \subseteq S$, $\bigcup_i T_i \subseteq \overline{S}$.
- $\sum_i |S_i| \geq \mu^+(S)/4$.
- *Each S_i (and T_i) is contained in a level set.*
- *No vertex is present in more than $2r$ cover graphs G_{S_i,T_i}.*

The main upshot of this lemma is that one can "break up" the $(S, \overline{S})$ routing problem into a collection of (S_i, T_i) routing problems, where the S_i, T_i are level subsets. Moreover, the interaction between the various cover graphs is limited, because of the last bullet point. Given that the separation distance of S is r, we believe that for a constant fraction (by total size) of the matched pairs $(S_i, T_i; \phi_i)$, the distance of these pairs is $\Omega(r)$. If Conjecture 3 is true, we can route $\Omega(r|S_i|)$ units of edge disjoint flow with vertex congestion r. Any vertex participates is at most $2r$ such flow. We had hoped to overlay such flows and get an overall vertex congestion of $O(r^2)$. Unfortunately, a direct overlay of flows leads to an edge congestion of $O(r)$, which is not useful for Conjecture 26. Nonetheless, it felt that a proof of Conjecture 3 with the proof techniques of Theorem 24 might yield insight into Conjecture 26.

References

1. Nir Ailon and Bernard Chazelle. Information theory in property testing and monotonicity testing in higher dimension. *Information and Computation*, 204(11):1704–1717, 2006. doi:10.1016/J.IC.2006.06.001.
2. Nir Ailon, Bernard Chazelle, Seshadhri Comandur, and Ding Liu. Estimating the distance to a monotone function. *Random Structures Algorithms*, 31(3):371–383, 2007. Prelim. version in Proc., RANDOM 2004. doi:10.1002/RSA.20167.
3. Paul Bastide, Carla Groenland, Hugo Jacob, and Tom Johnston. Exact antichain saturation numbers via a generalisation of a result of lehman-ron. *Combinatorial Theory*, 4(1), 2024. doi:10.5070/C64163848.
4. Aleksandrs Belovs and Eric Blais. A polynomial lower bound for testing monotonicity. *SIAM Journal on Computing (SICOMP)*, 50(3):406–433, 2021. Prelim. version in Proc., STOC 2016. doi:10.1137/16M1097006.
5. Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. L_p-testing. In *Proceedings, ACM Symposium on Theory of Computing (STOC)*, 2014.
6. Arnab Bhattacharyya. A note on the distance to monotonicity of boolean functions. Technical Report 012, Electronic Colloquium on Computational Complexity (ECCC), 2008.
7. Arnab Bhattacharyya, Elena Grigorescu, Madhav Jha, Kyoming Jung, Sofya Raskhodnikova, and David Woodruff. Lower bounds for local monotonicity reconstruction from transitive-closure spanners. *SIAM Journal on Discrete Mathematics (SIDMA)*, 26(2):618–646, 2012. Prelim. version in Proc., RANDOM 2010. doi:10.1137/100808186.
8. Hadley Black, Deeparnab Chakrabarty, and C. Seshadhri. A $o(d) \cdot \text{polylog}(n)$ monotonicity tester for Boolean functions over the hypergrid $[n]^d$. In *Proceedings, ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2018.
9. Hadley Black, Deeparnab Chakrabarty, and C. Seshadhri. Domain reduction: A $o(d)$ tester for boolean functions in d-dimensions. In *Proceedings, ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2020.
10. Hadley Black, Deeparnab Chakrabarty, and C. Seshadhri. A $d^{1/2+o(1)}$ monotonicity tester for boolean functions on d-dimensional hypergrids. In *Proceedings, IEEE Symposium on Foundations of Computer Science (FOCS)*, 2023.
11. Hadley Black, Deeparnab Chakrabarty, and C. Seshadhri. Directed isoperimetric theorems for boolean functions on the hypergrid and an $\widetilde{O}(n\sqrt{d})$ monotonicity tester. In *Proceedings, ACM Symposium on Theory of Computing (STOC)*, 2023.

12 Hadley Black, Iden Kalemaj, and Sofya Raskhodnikova. Isoperimetric inequalities for real-valued functions with applications to monotonicity testing. *arXiv*, abs/2011.09441, 2020. arXiv:2011.09441.

13 Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. *Computational Complexity*, 21(2):311–358, 2012. Prelim. version in Proc., CCC 2011. doi:10.1007/S00037-012-0040-X.

14 Eric Blais, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Lower bounds for testing properties of functions over hypergrid domains. In *Proceedings, IEEE Conference on Computational Complexity (CCC)*, 2014. doi:10.1109/CCC.2014.38.

15 Mark Braverman, Subhash Khot, Guy Kindler, and Dor Minzer. Improved monotonicity testers via hypercube embeddings. In *Innovations in Theoretical Computer Science (ITCS)*, pages 25:1–25:24, 2023. doi:10.4230/LIPICS.ITCS.2023.25.

16 Jop Briët, Sourav Chakraborty, David García Soriano, and Ari Matsliah. Monotonicity testing and shortest-path routing on the cube. *Combinatorica*, 32(1):35–53, 2012. doi:10.1007/S00493-012-2765-1.

17 Deeparnab Chakrabarty, Kashyap Dixit, Madhav Jha, and C Seshadhri. Property testing on product distributions: Optimal testers for bounded derivative properties. *ACM Trans. on Algorithms (TALG)*, 13(2):1–30, 2017. Prelim. version in Proc., SODA 2015. doi:10.1145/3039241.

18 Deeparnab Chakrabarty and C. Seshadhri. Optimal bounds for monotonicity and Lipschitz testing over hypercubes and hypergrids. In *Proceedings, ACM Symposium on Theory of Computing (STOC)*, 2013. doi:10.1145/2488608.2488661.

19 Deeparnab Chakrabarty and C. Seshadhri. An $o(n)$ monotonicity tester for Boolean functions over the hypercube. *SIAM Journal on Computing (SICOMP)*, 45(2):461–472, 2014. Prelim. version in Proc., STOC 2013. doi:10.1137/13092770X.

20 Xi Chen, Anindya De, Rocco A. Servedio, and Li-Yang Tan. Boolean function monotonicity testing requires (almost) $O(n^{1/2})$ non-adaptive queries. In *Proceedings, ACM Symposium on Theory of Computing (STOC)*, 2015.

21 Xi Chen, Rocco A. Servedio, and Li-Yang. Tan. New algorithms and lower bounds for monotonicity testing. In *Proceedings, IEEE Symposium on Foundations of Computer Science (FOCS)*, 2014.

22 Xi Chen, Erik Waingarten, and Jinyu Xie. Beyond Talagrand: New lower bounds for testing monotonicity and unateness. In *Proceedings, ACM Symposium on Theory of Computing (STOC)*, 2017.

23 William Cook, William Cunningham, William Pulleybank, and Alexander Schrijver. *Combinatorial Optimization*. Wiley Interscience Series, 1998.

24 Yevgeny Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. *Proceedings, International Workshop on Randomization and Computation (RANDOM)*, 1999.

25 Funda Ergun, Sampath Kannan, Ravi Kumar, Ronitt Rubinfeld, and Mahesh Viswanathan. Spot-checkers. *J. Comput. System Sci.*, 60(3):717–751, 2000. Prelim. version in Proc., STOC 1998. doi:10.1006/JCSS.1999.1692.

26 Shahar Fattal and Dana Ron. Approximating the distance to monotonicity in high dimensions. *ACM Trans. on Algorithms (TALG)*, 6(3), 2010. doi:10.1145/1798596.1798605.

27 Yuval Filmus, Hamed Hatami, Steven Heilman, Elchanan Mossel, Ryan O'Donnell, Sushant Sachdeva, Andrew Wan, and Karl Wimmer. Real analysis in computer science: A collection of open problems. https://simons.berkeley.edu/sites/default/files/openprobsmerged.pdf, 2014.

28 Eldar Fischer. On the strength of comparisons in property testing. *Information and Computation*, 189(1):107–116, 2004. doi:10.1016/J.IC.2003.09.003.

29 Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, and Ronitt Rubinfeld. Monotonicity testing over general poset domains. *Proceedings, ACM Symposium on Theory of Computing (STOC)*, 2002. doi:10.1145/509907.509977.

30 O. Goldreich, S. Goldwasser, and S. Ron. A note of testing monotonicity, 1997. Technical report. URL: https://www.wisdom.weizmann.ac.il/~oded/p_testMON.html.

31 Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samordinsky. Testing monotonicity. *Combinatorica*, 20:301–337, 2000. Prelim. version in Proc., FOCS 1998, with authors Goldreich, Goldwasser, Lehman, and Ron. doi:10.1007/S004930070011.

32 Shirley Halevy and Eyal Kushilevitz. Distribution-free property testing. *Proceedings, International Workshop on Randomization and Computation (RANDOM)*, 2003.

33 Shirley Halevy and Eyal Kushilevitz. Testing monotonicity over graph products. *Random Structures Algorithms*, 33(1):44–67, 2008. Prelim. version in Proc., ICALP 2004. doi:10.1002/RSA.20211.

34 Nathaniel Harms and Yuichi Yoshida. Downsampling for testing and learning in product distributions. In *Proceedings, International Colloquium on Automata, Languages and Programming (ICALP)*, volume 229, pages 71:1–71:19, 2022. doi:10.4230/LIPICS.ICALP.2022.71.

35 Subhash Khot, Dor Minzer, and Muli Safra. On monotonicity testing and boolean isoperimetric-type theorems. *SIAM Journal on Computing*, 47(6):2238–2276, 2018. Prelim. version in Proc., FOCS 2015. doi:10.1137/16M1065872.

36 Eric Lehman and Dana Ron. On disjoint chains of subsets. *Journal of Combinatorial Theory, Series A*, 94(2):399–404, 2001. doi:10.1006/JCTA.2000.3148.

37 Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and Erik Waingarten. Approximating the distance to monotonicity of boolean functions. *Random Structures Algorithms*, 60(2):233–260, 2022. Prelim. version in Proc., SODA 2020. doi:10.1002/RSA.21029.

38 Sofya Raskhodnikova. Monotonicity testing. *Masters Thesis, MIT*, 1999.

39 Dana Ron, Ronitt Rubinfeld, Muli Safra, Alex Samorodnitsky, and Omri Weinstein. Approximating the influence of monotone boolean functions in $O(\sqrt{n})$ query complexity. *ACM Trans. Comput. Theory*, 4(4):11:1–11:12, 2012. Prelim. version in Proc., RANDOM 2011. doi:10.1145/2382559.2382562.

40 Michael E. Saks and C. Seshadhri. Parallel monotonicity reconstruction. In *Proceedings, ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2008. URL: http://dl.acm.org/citation.cfm?id=1347082.1347187.

Unraveling Universally Closest Refinements via Symmetric Density Decomposition and Fisher Market Equilibrium

T-H. Hubert Chan
Department of Computer Science, University of Hong Kong, Hong Kong

Quan Xue
Department of Computer Science, University of Hong Kong, Hong Kong

Abstract

We investigate the closest distribution refinements problem, which involves a vertex-weighted bipartite graph as input, where the vertex weights on each side sum to 1 and represent a probability distribution. A refinement of one side's distribution is an edge distribution that corresponds to distributing the weight of each vertex from that side to its incident edges. The objective is to identify a pair of distribution refinements for both sides of the bipartite graph such that the two edge distributions are as close as possible with respect to a specific divergence notion. This problem is a generalization of transportation, in which the special case occurs when the two closest distributions are identical. The problem has recently emerged in the context of composing differentially oblivious mechanisms.

Our main result demonstrates that a universal refinement pair exists, which is simultaneously closest under all divergence notions that satisfy the data processing inequality. Since differential obliviousness can be examined using various divergence notions, such a universally closest refinement pair offers a powerful tool in relation to such applications.

We discover that this pair can be achieved via locally verifiable optimality conditions. Specifically, we observe that it is equivalent to the following problems, which have been traditionally studied in distinct research communities: (1) hypergraph density decomposition, and (2) symmetric Fisher Market equilibrium.

We adopt a symmetric perspective of hypergraph density decomposition, in which hyperedges and nodes play equivalent roles. This symmetric decomposition serves as a tool for deriving precise characterizations of optimal solutions for other problems and enables the application of algorithms from one problem to another. This connection allows existing algorithms for computing or approximating the Fisher market equilibrium to be adapted for all the aforementioned problems. For example, this approach allows the well-known iterative proportional response process to provide approximations for the corresponding problems with multiplicative error in distributed settings, whereas previously, only absolute error had been achieved in these contexts. Our study contributes to the understanding of various problems within a unified framework, which may serve as a foundation for connecting other problems in the future.

2012 ACM Subject Classification Mathematics of computing → Distribution functions; Theory of computation → Mathematical optimization

Keywords and phrases closest distribution refinements, density decomposition, Fisher market equilibrium

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.35

Related Version *Full Version*: https://arxiv.org/abs/2406.17964 [12]

Funding *T-H. Hubert Chan*: This research was partially supported by the Hong Kong RGC grants 17201823, 17203122 and 17202121.



© T-H. Hubert Chan and Quan Xue;
licensed under Creative Commons License CC-BY 4.0



1 Introduction

In this work, we explore several existing and new problems under the same unifying framework. By unraveling previously unknown connections between these seemingly unrelated problems, we offer deeper insights into them. Moreover, existing solutions for one problem can readily offer improvements to other problems. Input instances for these problems all have the following form.

Input Instance. An instance is a vertex-weighted bipartite graph with vertex bipartition $\mathcal{I} = \mathcal{I}^{(0)} \uplus \mathcal{I}^{(1)}$, edge set $\mathcal{F}$ and positive[1] vertex weights $w : \mathcal{I} \to \mathbb{R}_{>0}$. For some problems, it is convenient to normalize the vertex weights into distributions, i.e., the vertex weights on each side sum to 1; in this case, we call this a *distribution* instance. We use $\mathbb{B} := \{0, 1\}$ to index the two sides. Given a side $\iota \in \mathbb{B}$, we use $\bar{\iota} := \iota \oplus 1$ to denote the other side.

Solution Concept. We shall see that in each problem, *refinement* is the underlying solution concept. Given a side $\iota \in \mathbb{B}$, a refinement[2] $\alpha^{(\iota)}$ on vertex weights on side ι indicates how each vertex $i \in \mathcal{I}^{(\iota)}$ distributes its weight $w(i)$ among its incident edges in $\mathcal{F}$. In other words, $\alpha^{(\iota)}$ is a vector of edge weights in $\mathcal{F}$ such that for each $i \in \mathcal{I}^{(\iota)}$, the sum of weights of edges incident on i equals $w(i)$. While some problems may require refinements of vertex weights from one side only, it will still be insightful to consider a *refinement pair* $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$ of vertex weights on both sides.

The main problem we focus on is the closest distribution refinement problem, which we show is intricately related to other problems. In probability theory, a *divergence* notion quantifies how close two distributions are to each other, where a smaller quantity indicates that they are closer. Common examples of divergences include total variation distance and KL-divergence. Observe that a divergence notion D needs not be symmetric, i.e. for two distributions P and Q on the same sample space, it is possible that $\mathsf{D}(P\|Q) \neq \mathsf{D}(Q\|P)$.

▶ **Definition 1** (Closest Distribution Refinement Problem). *Given a distribution instance $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$ and a divergence notion D, the goal is to find a distribution refinement pair $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$ such that the distributions $\alpha^{(0)}$ and $\alpha^{(1)}$ on the edges $\mathcal{F}$ are as close to each other as possible under D, i.e., $\mathsf{D}(\alpha^{(0)}\|\alpha^{(1)})$ is minimized.*

Universally Closest Refinement Pair. It is not clear *a priori* if a refinement pair achieving the minimum for one divergence notion would also achieve the minimum for another divergence notion. For a non-symmetric divergence D, a pair $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$ minimizing $\mathsf{D}(\alpha^{(0)}\|\alpha^{(1)})$ may not necessarily achieve the minimum $\mathsf{D}(\alpha^{(1)}\|\alpha^{(0)})$. The special case when $\vec{\alpha}$ consists of a pair of identical refinements is also known as w-transportation or perfect fractional matching [6, 9, 27, 34], in which case $\vec{\alpha}$ is obviously the minimizer for all divergences.

In general, it would seem ambitious to request a *universally closest* refinement pair that simultaneously minimizes all "reasonable" divergence notions. In the literature, all commonly used divergence notions satisfy the *data processing inequality*, which intuitively says that the divergence of two distributions cannot increase after post-processing. Formally, for any deterministic function $\varphi : \Omega \to \widehat{\Omega}$, the induced distribution $\varphi(P)$ and $\varphi(Q)$ on $\widehat{\Omega}$ satisfy the inequality: $\mathsf{D}(\varphi(P)\|\varphi(Q)) \leq \mathsf{D}(P\|Q)$. Surprisingly, we show that such a universally closest pair exists.

[1] Vertices with zero weights are removed and identified with the empty set $\emptyset$.

[2] Since vertex weights on one side can be interpreted as a probability distribution, allocating the weight of each vertex to its incident edges can be viewed as a refinement of the original distribution.

▶ **Theorem 2** (Universally Closest Refinement Pair). *Given a distribution instance* $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$*, there exists a distribution refinement pair* $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$ *that minimizes* $\mathsf{D}(\alpha^{(0)} \| \alpha^{(1)})$ *simultaneously for all divergence notions* D *that satisfy the data processing inequality.*

Moreover, this pair can be returned using the near-linear time algorithm of Chen et al. [14] for the minimum quadratic-cost flow problem.

Since Definition 1 gives a clean mathematical problem, we first describe the technical contributions and approaches of this work, and we describe its motivation and application in Section 1.2.

While the technical proofs are given in Sections 4 and 5, we give some motivation for why the problem is studied in the first place and sketch the key ideas for our proofs.

1.1 Our Technical Contributions and Approaches

In addition to the problem in Definition 1, we shall see that the following solution properties are also the key to solving other problems as well.

Desirable Solution Properties. Two desirable solution properties are *local maximin* and *proportional response*.

- **Local Maximin.** A refinement $\alpha^{(\iota)}$ on vertex weights on side ι can also be interpreted as how each vertex in $\mathcal{I}^{(\iota)}$ allocates its weight to its neighbors on the other side $\mathcal{I}^{(\bar{\iota})}$ via the edges in $\mathcal{F}$. Specifically, for $i \in \mathcal{I}^{(\iota)}$ and $j \in \mathcal{I}^{(\bar{\iota})}$, $\alpha^{(\iota)}(ij)$ is the *payload* received by j from i, where we rely on the superscript in $\alpha^{(\iota)}$ to indicate that the payload is sent from side ι to side $\bar{\iota}$ along an edge in $\mathcal{F}$. Consequently, $\alpha^{(\iota)}$ induces a payload vector $p \in \mathbb{R}^{\mathcal{I}^{(\bar{\iota})}}$, where $p(j) = \sum_i \alpha^{(\iota)}(ij)$ is the total payload received by j in the allocation $\alpha^{(\iota)}$. Recalling that vertex j also has a weight, the payload density (resulting from $\alpha^{(\iota)}$) of j is $\rho(j) := \frac{p(j)}{w(j)}$. The refinement $\alpha^{(\iota)}$ is *locally maximin* if for each $i \in \mathcal{I}^{(\iota)}$, vertex i distributes a non-zero payload $\alpha^{(\iota)}(ij) > 0$ to vertex j only if j is among the neighbors of i with minimum payload densities (resulting from $\alpha^{(\iota)}$).
  The intuition of the term "maximin" is that vertex i tries to maximize the minimum payload density among its neighbors by giving its weight to only neighbors that achieve the minimum payload density.
- **Proportional Response.** Given a refinement $\alpha^{(\iota)}$ on vertex weights on side ι, its proportional response $\alpha^{(\bar{\iota})}$ is a refinement on vertex weights from the other side $\bar{\iota}$. Specifically, vertex $j \in \mathcal{I}^{(\bar{\iota})}$ distributes its weight $w(j)$ proportionately according to its received payloads: $\alpha^{(\bar{\iota})}(ij) = \frac{\alpha^{(\iota)}(ij)}{p(j)} \cdot w(j)$.

Several problems in the literature studied in different contexts can be expressed with the same goal.

▶ **Goal 3** (High-Level Goal). Given an instance $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$, find a refinement pair $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$ such that both refinements are locally maximin and proportional response to each other.

This abstract problem have been studied extensively in the literature under different contexts, and various algorithms to exactly achieve or *approximate* (whose meaning we will explain) Goal 3 have been designed. One contribution of this work is to unify all these problems under our abstract framework. Consequently, we readily offer new approximation results for some problems that are unknown before this connection is discovered.

We next describe how our abstract framework is related to each considered problem and what new insights or results are achieved, which also provide the tools to solve or approximate the problem in Definition 1.

1.1.1 Hypergraph Density Decomposition

An instance $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$ is interpreted as a *hypergraph* H. Each element in $i \in \mathcal{I}^{(0)}$ is a hyperedge and each element in $j \in \mathcal{I}^{(1)}$ is a node, where $\{i, j\} \in \mathcal{F}$ means that node j is contained in hyperedge i. Observe that in this hypergraph, both the hyperedges and the nodes are weighted.

Densest Subset Problem. Given a subset $S \subseteq \mathcal{I}^{(1)}$ of nodes in a (weighted) hypergraph, its density is $\rho(S) := \frac{w(H[S])}{w(S)}$, where $H[S]$ is the collection of hyperedges containing only nodes inside S. The densest subset problem requests a subset S with the maximum density.

Density Decomposition. The decomposition can be described by an iterative process that generates a density vector $\rho_* \in \mathbb{R}^{\mathcal{I}^{(1)}}$ for the nodes in the hypergraph by repeated peeling off densest subsets. First, the (unique) maximal densest subset S_1 is obtained, and each $j \in S_1$ will be assigned the density $\rho_*(j) = \frac{w(H[S_1])}{w(S_1)}$ of S_1. Then, the hyperedges $H[S_1]$ and the subset S_1 are removed from the instance; note that any hyperedge containing a node outside S_1 will remain. The process is applied to the remaining instance until eventually every node in $\mathcal{I}^{(1)}$ is removed and assigned its density. The collection of subsets found in the iterative process forms the density decomposition $\mathcal{I}^{(1)} = \uplus_{k \geq 1} S_k$.

As mentioned in [22], the earliest variant of this decomposition was introduced by Fujishige [19] in the context of polymatroids. We describe the related work in more detail in Section 1.3.

Connection with the Local Maximin Condition. Charikar [13] gave an LP (that can be easily generalized to weighted hypergraphs) which solves the densest subset problem exactly, and a refinement $\alpha^{(0)}$ on the weights of hyperedges $\mathcal{I}^{(0)}$ corresponds exactly to a feasible solution of the dual LP. Replacing the linear function in this dual LP with a quadratic function while keeping the same feasible variable $\alpha^{(0)}$, Danisch et al. [15] considered a quadratic program (that has also appeared in [19]). They showed that a refinement corresponds to an optimal solution to the quadratic program *iff* it is locally maximin[3]. Moreover, a locally maximin refinement also induces the aforementioned density vector ρ_*.

As observed in [22], the quadratic program can be expressed as a minimum quadratic-cost flow problem, which can be solved optimally using the nearly-linear time algorithm of Chen et al. [14]. As we shall show that our considered problems are all equivalent to Goal 3, this implies the algorithmic result in Theorem 2.

Symmetry of Density Decomposition. We observe that the density decomposition is symmetric between the hyperedges and the nodes. Recall that in iteration k of the density decomposition, a subset S_k is produced, together with its corresponding collection $\widehat{S}_k$ of hyperedges that determines the density $\frac{w(\widehat{S}_k)}{w(S_k)}$. Therefore, in addition to a partition of nodes, there is a corresponding decomposition of the hyperedges $\mathcal{I}^{(0)} = \uplus_{k \geq 1} \widehat{S}_k$.

[3] Instead of using the term "maximin", the same concept is called "stability" in [15].

Given an abstract instance $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$, one could have interpreted it differently with $\mathcal{I}^{(0)}$ as nodes and $\mathcal{I}^{(1)}$ as hyperedges. When the density decomposition is constructed with the roles reversed, the exact sequence $\{(\widehat{S}_k, S_k)\}_k$ of pairs is obtained, but in reversed order. Intuitively, $(\widehat{S}_1, S_1)$ is the densest from the perspective of $S_1 \subseteq \mathcal{I}^{(1)}$, but it is the least dense from the perspective of $\widehat{S}_1 \subseteq \mathcal{I}^{(0)}$.

The symmetric nature of the density decomposition has been mentioned recently by Harb et al. [23, Theorem 13], but we shall explain later this fact may be inferred when different earlier works are viewed together. Hence, this observation may be considered as part of the folklore. Since we need some crucial properties of the symmetric density decomposition to prove Theorem 2, for the sake of completeness, the connection of Goal 3 and this folklore result is described in Section 3.

Approximation Notions. Even though the exact density decomposition can be achieved via LP or maximum flow, it can be computationally expensive in practice. In the literature, efficient approximation algorithms have been considered. While there are different attempts to define approximation notions for density decompositions, it is less clear what they mean when they are applied to different problems.

Observe that a refinement pair $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$ achieving Goal 3 may not be unique, but the induced payload vectors $(p_*^{(0)}, p_*^{(1)})$ and density vectors $(\rho_*^{(0)}, \rho_*^{(1)})$ are unique. When we say *approximating* Goal 3, we mean finding a refinement pair such that the induced payload and density vectors have some approximation guarantees. We shall see that approximation guarantees[4] on these vectors can be translated readily to approximation notions for various problems.

1.1.2 Unraveling Universally Closest Distribution Refinements via Symmetric Density Decomposition

Restricted Class of Divergence. As a first step towards proving Theorem 2, we consider a restricted class of divergences known as *hockey-stick* divergence. For each $\gamma \geq 0$,

$$\mathsf{D}_\gamma(P\|Q) := \sup_{S \subseteq \Omega} Q(S) - \gamma \cdot P(S).$$

The hockey-stick divergence is related to the well-known (ϵ, δ)-differential privacy inequality. Specifically, $\mathsf{D}_{e^\epsilon}(P\|Q) \leq \delta$ *iff* for all subsets $S \subseteq \Omega$,

$$Q(S) \leq e^\epsilon \cdot P(S) + \delta.$$

We first consider the universal closest distribution refinements with respect to the class of hockey-stick divergences $\mathcal{D}_{\text{HS}} := \{\mathsf{D}_\gamma : \gamma \geq 0\}$. In Lemma 31, we show that the restricted variant of Theorem 2 to the class $\mathcal{D}_{\text{HS}}$ is equivalent to the following problem.

Universal Refinement Matching Problem. Given an instance $G = (\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$ and two non-negative weights $\vec{c} = (c^{(0)}, c^{(1)}) \in \mathbb{R}^{\mathbb{B}}$, we use $G^{(\vec{c})}$ to denote the instance of (fractional) maximum matching on the bipartite graph $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F})$ with vertex capacity constraints. For side ι and $i \in \mathcal{I}^{(\iota)}$, the capacity constraint at vertex i is $c^{(\iota)} \cdot w(i)$. A fractional solution consists of assigning non-negative weights to the edges in $\mathcal{F}$ such that the weighted degree of every vertex does not exceed its capacity constraint. Observe that for any refinement pair $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$, the fractional matching $\vec{c} \bullet \vec{\alpha}$ defined below is a feasible solution to $G^{(\vec{c})}$:

$$\vec{c} \bullet \vec{\alpha}(ij) := \min\{c^{(0)} \cdot \alpha^{(0)}(ij), c^{(1)} \cdot \alpha^{(1)}(ij)\} \text{ for all } \{i, j\} \in \mathcal{F}.$$

[4] See Definition 12 for approximation notions of *muliplicative* vs *absolute* errors on vectors.

The goal is to find a refinement pair $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$ such that for any weight pair $\vec{c}$, the edge weights $\vec{c} \bullet \vec{\alpha}$ form an optimal (fractional) solution to the maximum matching instance $G^{(\vec{c})}$.

The structure of the symmetric density decomposition described in Section 1.1.1 makes it convenient to apply a primal-dual analysis to achieve the following in Section 4.

▶ **Theorem 4** (Universal Maximum Refinement Matching). *Given an instance $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$, suppose a distribution refinement pair $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$ satisfies Goal 3. Then, for any $\vec{c} = (c^{(0)}, c^{(1)})$, the (fractional) solution $\vec{c} \bullet \vec{\alpha}$ is optimal for the maximum matching instance $G^{(\vec{c})}$.*

Extension to General Case. To extend Theorem 2 beyond the restricted divergence class $\mathcal{D}_{\mathrm{HS}}$, we consider the closest distribution refinement problem under a more general probability distance notion.

Capturing All Data Processing Divergences via Power Functions. As opposed to using a single number by a divergence notion to measure the closeness of two distributions, the *power function* $\mathsf{Pow}(P\|Q) : [0,1] \to [0,1]$ between distributions P and Q captures enough information to recover all divergence notions that satisfy the data processing inequality. We will give a formal explanation in Section 5. A main result is given in Theorem 37, which states that any distribution refinement pair satisfying Goal 3 achieves the minimal power function, from which Theorem 2 follows immediately.

Approximation Results. In the full version [12], we show that a refinement that induces a payload with multiplicative error can lead to approximation guarantees for other problems. For the universal refinement matching problem, a natural approximation notion is based on the standard approximation ratio for the maximum matching problem.

On the other hand, achieving a meaningful approximation notion for the universally closest distribution refinement problem in terms of divergence values is more difficult. The reason is that any strictly increasing function on a data processing divergence still satisfies the data processing inequality. Therefore, any non-zero deviation from the correct answer may be magnified arbitrarily. Instead of giving approximation guarantees directly on the divergence values, we consider a technical notion of power function stretching, and the approximation quality of a refinement pair is quantified by how much the optimal power function is stretched. The formal statement is given in the full version [12].

1.1.3 Achieving Multiplicative Error via Proportional Response in Fisher Markets

We next interpret an instance $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$ as a special symmetric case of linear Fisher markets [18], which is itself a special case of the more general Arrow-Debreu markets [2]. This allows us to transfer any algorithmic results for Fisher/Arrow-Debreu market equilibrium to counterparts for Goal 3. Specifically, we can use a quick iterative procedure *proportional response* [39, 40] to approximate the densities of vertices in the density decomposition in Section 1.1.1 with multiplicative error, where only absolute error has been achieved before [15, 22]. In addition to being interesting in its own right, multiplicative error is also used to achieve the approximation results in Section 1.1.2.

Market Interpretation. Even though in an instance, the roles of the two sides are symmetric, it will be intuitive to think of $\mathcal{I}^{(0)}$ as buyers and $\mathcal{I}^{(1)}$ as sellers. For buyer $i \in \mathcal{I}^{(0)}$, the weight $w(i)$ is its *budget*. In the symmetric variant, the utilities of buyers have a special form.

Each seller $j \in \mathcal{I}^{(1)}$ has one divisible unit of good j for which a buyer i has a positive utility exactly when $\{i, j\} \in \mathcal{F}$, in which case the utility per unit of good j is $w(j)$ (that is common for all buyers that have a positive utility for j). For a general Fisher market, a buyer i can have an arbitrary utility $w_i(j)$ for one unit of good j.

Market Allocation and Equilibrium. For the symmetric variant, an allocation can be represented by a refinement pair $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$, in which $\alpha^{(1)}$ must be a proportional response to $\alpha^{(0)}$. For the refinement $\alpha^{(0)}$ on the buyers' budgets, the induced payload vector $p^{(1)}$ on the sellers corresponds to the *price* vector on goods. For the refinement $\alpha^{(1)}$ on the sellers' weights, the induced payload vector $p^{(0)}$ on the buyers corresponds to the utilities received by the buyers. It is well-known (explained in the full version [12]) that a market equilibrium is characterized by a locally maximin refinement $\alpha^{(0)}$ of buyers' budgets together with its proportional response $\alpha^{(1)}$. From our theory of symmetric density decomposition in Section 1.1.1, this is equivalent to Goal 3.

Alternative Characterization. The fact that the local maximin condition on the buyers' budget allocation is equivalent to Fisher market equilibrium readily allows us to borrow any algorithmic results for market equilibrium to solve or approximate Goal 3. In the full version [12], we give details on the following observations.

- In linear Fisher markets (not necessarily symmetric), we show that the local maximin condition on sellers' goods allocation to buyers can be used to characterize a market equilibrium. This is an alternative characterization of linear Fisher market equilibrium that involves only goods allocation and buyers' utilities, but not involving good prices.
- On the other hand, we show that there exists an Arrow-Debreu market such that there is a goods allocation satisfying the local maximin condition, but it is not an equilibrium.

1.2 Motivation and Application for Universally Closest Distribution Refinements

The problem in Definition 26 arises in the study of the composition of differentially oblivious mechanisms [41, 42], which is a generalization of the more well-known concept of differential privacy (DP) [17].

A Quick Introduction to Differential Privacy/Obliviousness. Consider a (randomized) mechanism $\mathcal{M}$ that takes an input from space $\mathcal{X}$ and returns an output from space $\mathcal{Y}$ that will be passed to a second mechanism. During the computation of $\mathcal{M}$, an adversary can observe some *view* from space $\mathcal{V}$ that might not necessarily include the output in general. Suppose at first we just consider the privacy/obliviousness of $\mathcal{M}$ on its own, which measures how much information the adversary can learn about the input from its view. Intuitively, it should be difficult for the adversary to distinguish between two similar inputs from their corresponding views produced by the mechanism. Formally, similar inputs in $\mathcal{X}$ are captured by a *neighboring* symmetric relation, and the privacy/obliviousness of the mechanism $\mathcal{M}$ requires that the two distributions of views produced by neighboring inputs are "close", where closeness can be quantified by some notion of divergence. As mentioned in Section 1.1.2, the popular notion of (ϵ, δ)-DP is captured using hockey-stick divergences.

Composition of Differentially Oblivious Mechanisms. Suppose the output of mechanism $\mathcal{M}$ from space $\mathcal{Y}$ is passed as an input to a second mechanism $\mathcal{M}'$, which is differentially oblivious with respect to some neighboring relation on $\mathcal{Y}$. Hence, given input x to $\mathcal{M}$, one should consider the joint distribution $\mathcal{M}(x)$ on the space $\mathcal{V} \times \mathcal{Y}$ of views and outputs.

The notion of neighbor-preserving differential obliviousness (NPDO) has been introduced [41, 42] to give sufficient conditions on $\mathcal{M}$ that allows composition with such a mechanism $\mathcal{M}'$ so that the overall composition preserves differential obliviousness.

Specifically, given neighboring inputs $x_0 \sim x_1$, conditions are specified on the two distributions $\mathcal{M}(x_0)$ and $\mathcal{M}(x_1)$. Unlike the case for differential privacy/obliviousness, it is not appropriate to just require that the two distributions are close. This is because the adversary can only observe the view component of $\mathcal{V} \times \mathcal{Y}$. Moreover, one also needs to incorporate the neighboring relation on the output space $\mathcal{Y}$, which is needed for the privacy guarantee for the second mechanism $\mathcal{M}'$.

The definition of NPDO is more conveniently described via the terminology offered by our abstract instance $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$. Here, the sets $\mathcal{I}^{(0)}$ and $\mathcal{I}^{(1)}$ are two copies of $\mathcal{V} \times \mathcal{Y}$. For each $\iota \in \mathbb{B}$, the weights w on $\mathcal{I}^{(\iota)}$ corresponds to the distribution $\mathcal{M}(x_\iota)$. Finally, $(v_0, y_0) \in \mathcal{I}^{(0)}$ and $(v_1, y_1) \in \mathcal{I}^{(1)}$ are neighbors in $\mathcal{F}$ *iff* $v_0 = v_1$ and $y_0 \sim y_1$ are neighboring in $\mathcal{Y}$.

In the original definition [42], the mechanism $\mathcal{M}$ is said to be d-NPDO with respect to divergence D if for the constructed instance $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$, there exists a distribution refinement pair $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$, which may depend on the divergence D, such that $\mathsf{D}(\alpha^{(0)} \| \alpha^{(1)}) \le d$. Examples of considered divergences include the standard hockey-stick divergence and Rényi divergence [31].

Since it only makes sense to consider divergences satisfying the data processing inequality in the privacy context, the universally closest refinement pair guaranteed in Theorem 2 makes the notion of NPDO conceptually simpler, because the closest refinement pair does not need to specifically depend on the considered divergence.

We believe that this simplification will be a very useful tool. In fact, to achieve advanced composition in [42], a very complicated argument has been used to prove the following fact, which is just an easy corollary of Theorem 2.

▶ **Fact 5** (Lemma B.1 in [42]). *Given a distribution instance $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$, suppose there exist two refinement pairs $\vec{\alpha}_s = (\alpha_s^{(0)}, \alpha_s^{(1)})$ and $\vec{\alpha}_t = (\alpha_t^{(0)}, \alpha_t^{(1)})$ such that the hockey-stick divergences $\mathsf{D}_\gamma(\alpha_s^{(0)} \| \alpha_s^{(1)})$ and $\mathsf{D}_\gamma(\alpha_t^{(1)} \| \alpha_t^{(0)})$ are both at most δ.*

Then, there exists a single refinement pair $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$ such that both $\mathsf{D}_\gamma(\alpha^{(0)} \| \alpha^{(1)})$ and $\mathsf{D}_\gamma(\alpha^{(1)} \| \alpha^{(0)})$ are at most δ.

1.3 Related Work

The densest subset problem is a classical problem in combinatorial optimization with numerous real-world applications in data mining, network analysis, and machine learning [1, 13, 21, 26, 29, 30, 35], and it is the basic subroutine in defining the density decomposition.

Density Decomposition. As mentioned in Section 1.1.1, the earliest reference to the density decomposition and its associated quadratic program were introduced by Fujishige [19]. Even though this decomposition was defined for the more general submodular or supermodular case, the procedure has been rediscovered independently many times in the literature for the special bipartite graph case (or equivalently its hypergraph interpretation).

Earlier we described the decomposition procedure by repeatedly peeling off maximal densest subsets in a hypergraph. This was the approach by Tatti and Gionis [36, 37] and Danisch et al. [15] . Moreover, the same procedure has been used to define hypergraph Laplacians in the context of the spectral properties of hypergraphs [11].

In the literature, there is another line of research that defines the decomposition by repeatedly peeling off least densest subsets instead. However, given a subset $S \subseteq V$ of nodes in a weighted hypergraph $H = (V, E)$, a different notion of density is considered: $\mathfrak{r}(S) := \frac{w(E(S))}{w(S)}$, where $E(S)$ is the collection of hyperedges that have non-empty intersection with S. The difference is that before we consider $E[S]$ that is the collection of hyperedges totally contained inside S. Then, an alternative procedure can be defined by peeling off a maximal least densest subset under $\mathfrak{r}$ and performing recursion on the remaining graph.

This was the approach used by Goel et al. [20] to consider the communication and streaming complexity of maximum bipartite matching. The same procedure has been used by Bernstein et al. [7] to analyze the replacement cost of online maximum bipartite matching maintenance. Lee and Singla [28] also used this decomposition to investigate the the batch arrival model of online bipartite matching. Bansal and Cohen [4] have independently discovered this decomposition procedure to consider maximin allocation of hyperedge weights to incident nodes.

Symmetric Nature of Density Decomposition. At this point, the reader may ask how we know that the two procedures will give the same decomposition (but in reversed order). The answer is that exactly the same quadratic program has been used in [15] and [7] to analyze the corresponding approaches. This observation has also been made recently by Harb et al. [23, Theorem 13]. It is not too difficult to see that the least densest subset using the modified density $\mathfrak{r}$ is equivalent to the densest subset with the roles of vertices and hyperedges reversed. Therefore, Theorem 20 can also be obtained indirectly through these observations.

Connection between Density Decomposition and Fisher Market Equilibrium. Even before this work in which we highlight the local maximin condition, density decomposition has already surreptitiously appeared in algorithms computing Fisher market equilibrium. Jain and Vazirani [24] considered submodular utility allocation (SUA) markets, which generalize linear Fisher markets. Their procedure [24, Algorithm 9] for computing an SUA market equilibrium can be viewed as a masqueraded variant of the density decomposition by Fujishige [19]. Buchbinder et al. [10] used a variant of SUA market equilibrium to analyze online maintenance of matroid intersections, where equilibrium prices are also obtained by the density decomposition procedure.

In this work, we observe conversely that algorithms for approximating market equilibrium can also be directly used for approximating density decompositions. For the simple special case in which the input graph contains a perfect fractional matching, Motwani et al. [32] showed that several randomized balls-into-bins procedures attempting to maintain the locally maximin condition can achieve $(1 + \epsilon)$-approximation.

Distributed Iterative Approximation Algorithms. Even though efficient exact algorithms are known to achieve Goal 3, distributed iterative approximation algorithms have been considered for practical scenarios.

Density Decomposition. By applying the first-order iterative approach of Frank-Wolfe to the quadratic program, Danisch et al. [15] showed that T iterations, each of which takes $O(|\mathcal{F}|)$ work and may be parallelized, can achieve an absolute error of $O(\frac{1}{\sqrt{T}})$ on the density vector ρ_*. Harb et al. [23] showed that the method Greedy++ can be viewed as a noisy variant of Frank-Wolfe and hence has a similar convergence result.

By using a first-order method augmented with Nesterov momentum [33] such as accelerated FISTA [5], Elfarouk et al. [22] readily concluded that T iterations can achieve an absolute error of $O(\frac{1}{T})$. However, observe that if there is a vertex i such that $\rho_*(i) \leq \epsilon$, an absolute error of ϵ may produce an estimate $\rho(i)$ that is arbitrarily small.

Inspired by estimates of ρ_*, several notions of approximate density decompositions have been considered in previous works [15,22,37]. However, it is not clear if there is any meaningful interpretations of these approximate decompositions in the context of closest distribution refinements or market equilibrium.

Fisher Market Equilibrium. Market dynamics have been considered to investigate how agents' mutual interaction can lead to convergence to a market equilibrium. The proportional response (PR, *aka* tit-for-tat) protocol has been used by Wu and Zhang [39] to analyze bandwidth trading in a peer-to-peer network $G = (V, E)$ with vertex weights w. This can be seen as a special case of our instance $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$, where $\mathcal{I}^{(0)}$ and $\mathcal{I}^{(1)}$ are two copies of V, and $i \in \mathcal{I}^{(0)}$ and $j \in \mathcal{I}^{(1)}$ are neighbors in $\mathcal{F}$ *iff* the corresponding pair $\{i, j\} \in E$ are neighbors in G. Interestingly, they consider a *bottleneck* decomposition of the graph G which can be recovered from our symmetric density decomposition. They showed that the PR protocol converges to some equilibrium under this special model.

Subsequently, Zhang [40] showed that the PR protocol can be applied in Fisher markets between buyers and sellers, and analyzed the convergence rate to the market equilibrium in terms of multiplicative error. Birnbaum et al. [8] later improved the analysis of the PR protocol by demonstrating that it can be viewed as a generalized gradient descent algorithm.

1.4 Potential Impacts and Paper Organization

This work unifies diverse problems under a single framework, revealing connections and providing new approximation results. Future research directions are abundant. Examining the implications of various market models or equilibrium concepts on the densest subset problem and density decomposition opens the door for further exploration of the relationships between economic markets and combinatorial optimization problems. This line of inquiry could also lead to new insights into economics, particularly regarding market participants' behavior and the efficiency of different market mechanisms.

Paper Organization. In Section 2, we introduce the precise notation that will be used throughout the paper. In Section 3, we summarize the structural results regarding the symmetric density decomposition, which is used as a technical tool to analyze other problems. As mentioned in the introduction, Section 4 investigates the universal refinement matching problem which is the precursor for tackling the more challenging universally closest distribution refinements problem in Section 5. In Section 5.1, we first consider the special case of hockey-stick divergences; by considering power functions, we prove Theorem 2 in Section 5.2.

Additional Materials in the Full Version. In the full version [12], we present approximation results for the universal closest refinements problem by defining a stretching operation on power functions. Moreover, we recap the connection between the symmetric density decomposition and the symmetric Fisher market equilibrium. We give an alternative characterization of the Fisher market equilibrium in terms of the local maximin condition on sellers' goods allocation to buyers. Furthermore, we give details on the convergence rates of how the iterative proportional response protocol can achieve an approximation with multiplicative error for Goal 3.

2 Preliminaries

We shall consider several problems, whose inputs are in the form of vertex-weighted bipartite graphs.

▶ **Definition 6** (Input Instance). *An instance $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$ of the input consists of the following:*

- *Two disjoint vertex sets $\mathcal{I}^{(0)}$ and $\mathcal{I}^{(1)}$.*
  For the side $\iota \in \mathbb{B} := \{0,1\}$, we use $\bar{\iota} := \iota \oplus 1$ to indicate the other side different from ι.
- *A collection $\mathcal{F}$ of unordered pairs of vertices that is interpreted as a bipartite graph between vertex sets $\mathcal{I}^{(0)}$ and $\mathcal{I}^{(1)}$; in other words, if $f = \{i, j\} \in \mathcal{F}$, then exactly one element of f is in $\mathcal{I}^{(0)}$ and the other is in $\mathcal{I}^{(1)}$. In this case, we also denote $i \sim_{\mathcal{F}} j$ or just $i \sim j$.*
  A vertex is isolated if it has no neighbor.
- *Positive*[5] *vertex weights $w : \mathcal{I}^{(0)} \cup \mathcal{I}^{(1)} \to \mathbb{R}_{>0}$; for a vertex i, we use w_i or $w(i)$ to denote its weight.*
  For $\iota \in \mathbb{B}$, we also use $w^{(\iota)} \in \mathbb{R}^{\mathcal{I}^{(\iota)}}$ to denote the weight vector for each side.

  This notation highlights the symmetry between the two sets $\mathcal{I}^{(0)}$ and $\mathcal{I}^{(1)}$.

▶ Remark 7. In this work, we focus on the case that $\mathcal{I}^{(0)}$ and $\mathcal{I}^{(1)}$ are finite. Even though it is possible to extend the results to continuous and uncountable sets (which are relevant when those sets are interpreted as continuous probability spaces), the mathematical notations and tools involved are outside the scope of the typical computer science community.

▶ **Definition 8** (Allocation Refinement and Payload). *Given an input instance as in Definition 6, for side $\iota \in \mathbb{B}$, an allocation refinement (or simply refinement) $\alpha^{(\iota)} \in \mathbb{R}^{\mathcal{F}}_{\geq 0}$ of $w^{(\iota)} \in \mathbb{R}^{\mathcal{I}^{(\iota)}}$ can be interpreted as edge weights on $\mathcal{F}$ such that for each non-isolated vertex $i \in \mathcal{I}^{(\iota)}$,*

$$w^{(\iota)}(i) = \sum_{f \in \mathcal{F} : i \in f} \alpha^{(\iota)}(f).$$

In other words, if a vertex i has at least one neighbor, it distributes its weight w_i among its neighboring edges $\{i, j\}$ such that $i \sim j$. These edge weights due to the refinement $\alpha^{(\iota)}$ can be interpreted as received by vertices on the other side $\mathcal{I}^{(\bar{\iota})}$; this induces a payload vector $p^{(\bar{\iota})} \in \mathbb{R}^{\mathcal{I}^{(\bar{\iota})}}$ defined by:

$$\forall j \in \mathcal{I}^{(\bar{\iota})}, p^{(\bar{\iota})}(j) := \sum_{f : j \in f} \alpha^{(\iota)}(f).$$

With respect to the weights of vertices in $\mathcal{I}^{(\bar{\iota})}$, the (payload) density $\rho^{(\bar{\iota})} \in \mathbb{R}^{\mathcal{I}^{(\bar{\iota})}}$ is defined by:

$$\rho^{(\bar{\iota})}(j) := \frac{p^{(\bar{\iota})}(j)}{w(j)}.$$

When there is no risk of ambiguity, the superscripts (ι) and $(\bar{\iota})$ may be omitted.

▶ **Definition 9** (Proportional Response). *Suppose for side $\iota \in \mathbb{B}$, $\alpha^{(\iota)}$ is a refinement of the vertex weights on $\mathcal{I}^{(\iota)}$. Then, a proportional response to $\alpha^{(\iota)}$ is a refinement $\alpha^{(\bar{\iota})}$ of the vertex weights on the other side $\mathcal{I}^{(\bar{\iota})}$ satisfying the following:*

[5] Vertices with zero weight do not play any role and may be excluded.

- *If $j \in \mathcal{I}^{(\bar{\iota})}$ receives a positive payload $p^{(\bar{\iota})}(j) > 0$ from $\alpha^{(\iota)}$, then for each $i \sim j$, $\alpha^{(\bar{\iota})}(ij) = \frac{\alpha^{(\iota)}(ij)}{p^{(\bar{\iota})}(j)} \cdot w(j) = \frac{\alpha^{(\iota)}(ij)}{\rho^{(\bar{\iota})}(j)}$.*
- *If a non-isolated vertex $j \in \mathcal{I}^{(\bar{\iota})}$ receives zero payload from $\alpha^{(\iota)}$, then in $\alpha^{(\bar{\iota})}$, vertex j may distributes its weight $w(j)$ arbitrarily among its incident edges in $\mathcal{F}$.*

▶ **Definition 10** (Local Maximin Condition). *Given an input instance as in Definition 6, for $\iota \in \mathbb{B}$, suppose a refinement $\alpha^{(\iota)} \in \mathbb{R}_+^{\mathcal{F}}$ of $w^{(\iota)} \in \mathbb{R}^{\mathcal{I}^{(\iota)}}$ induces a payload density vector $\rho^{(\bar{\iota})} \in \mathbb{R}^{\mathcal{I}^{(\bar{\iota})}}$ as in Definition 8. Then, $\alpha^{(\iota)}$ is locally maximin if,*

$$\forall i \in \mathcal{I}^{(\iota)}, \quad \alpha^{(\iota)}(ij) > 0 \implies j \in \arg\min_{\ell \in \mathcal{I}^{(\bar{\iota})} : \ell \sim i} \rho^{(\bar{\iota})}(\ell).$$

In other words, the local maximin condition states that each vertex $i \in \mathcal{I}^{(\iota)}$ attempts to send a positive weight to a neighbor $j \sim i$ only if j achieves the minimum $\rho^{(\bar{\iota})}(\ell)$ among the neighbors ℓ of i.

▶ Remark 11. Observe that if $\alpha^{(\iota)}$ satisfies the local maximin condition, then every non-isolated vertex in $\mathcal{I}^{(\bar{\iota})}$ must receive a positive payload. This means that its proportional response $\alpha^{(\bar{\iota})}$ is uniquely determined.

▶ **Definition 12** (Notions of Error). *Given a vector ρ, we consider two notions of error for an approximation vector $\widehat{\rho}$.*

- Absolute error ϵ with respect to norm $\|\cdot\|$. *We have:* $\|\rho - \widehat{\rho}\| \le \epsilon$. *A common example is the standard ℓ_2-norm.*
- Multiplicative error ϵ. *For all coordinates i,* $|\rho(i) - \widehat{\rho}(i)| \le \epsilon \cdot |\rho(i)|$.

3 Folklore: Symmetric Density Decomposition

An instance $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$, as in Definition 6, may be interpreted as a hypergraph H. We have the vertex set $V = \mathcal{I}^{(1)}$ (*aka* the *ground* set) and the hyperedge set $E = \mathcal{I}^{(0)}$ (*aka* the *auxiliary* set), where $v \in V$ and $e \in E$ are neighbors in $\mathcal{F}$ *iff* the hyperedge e contains the vertex v. Moreover, w assigns weights to both V and E in the hypergraph.

Notation: Ground and Auxiliary Sets. For side $\iota \in \mathbb{B}$, the *open neighborhood* of a non-empty subset $S \subseteq \mathcal{I}^{(\iota)}$ consists of vertices on the other side that have at least one neighbor in S, i.e., $\mathcal{F}(S) := \{j \in \mathcal{I}^{(\bar{\iota})} : \exists i \in S, i \sim j\}$. Its *closed neighborhood* is $\mathcal{F}[S] := \{j \in \mathcal{F}(S) : \mathcal{F}(\{j\}) \subseteq S\}$, i.e., a vertex j is in $\mathcal{F}[S]$ *iff* it has at least one neighbor and all its neighbors are contained in S. Its density is defined as $\rho^{(\iota)}(S) := \frac{w(\mathcal{F}[S])}{w(S)}$, where the superscript ι may be omitted when the context is clear. When we consider the densities of subsets in $\mathcal{I}^{(\iota)}$, the set $\mathcal{I}^{(\iota)}$ plays the role of the *ground set* and the set $\mathcal{I}^{(\bar{\iota})}$ plays the role of the *auxiliary set*. Observe that this definition is consistent with the hypergraph (V, H) interpretation when $\iota = 1$ plays the role of the vertex set V.

Extension to Empty Set. In some of our applications, it makes sense to have isolated vertices in the input instance. For $\emptyset \subseteq \mathcal{I}^{(\iota)}$, we extend $\mathcal{F}(\emptyset) = \mathcal{F}[\emptyset] := \{j \in \mathcal{I}^{(\bar{\iota})} : \forall i \in \mathcal{I}^{(\iota)}, j \not\sim i\}$. For the purpose of defining $\rho(\emptyset)$, we use the convention that $\frac{0}{0} = 0$ and $\frac{x}{0} = +\infty$ for $x > 0$.

▶ **Definition 13** (Densest Subset). *When $\mathcal{I}^{(\iota)}$ plays the role of the ground set, a subset $S \subseteq \mathcal{I}^{(\iota)}$ is a densest subset, if it attains* $\max_{S \subseteq \mathcal{I}^{(\iota)}} \rho^{(\iota)}(S)$.

The following well-known fact characterizes a property of maximal densest subset (see [3, Lemma 4.1] for a proof).

▶ **Fact 14.** *With respect to set inclusion, the maximal densest subset in $\mathcal{I}^{(\iota)}$ is unique and contains all densest subsets.*

▶ **Definition 15** (Sub-Instance). *Given an instance $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$, the subsets $S^{(0)} \subseteq \mathcal{I}^{(0)}$ and $S^{(1)} \subseteq \mathcal{I}^{(1)}$ naturally induces a sub-instance $(S^{(0)}, S^{(1)}; \mathcal{F}'; w')$, where $\mathcal{F}'$ and w' are restrictions of $\mathcal{F}$ and w, respectively, to $S^{(0)} \cup S^{(1)}$.*

The following definition is essentially the same as the density decomposition of a hypergraph [36, 37].

▶ **Definition 16** (Density Decomposition with Ground Set). *Given an instance $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$, suppose for $\iota \in \mathbb{B}$, we consider $\mathcal{I}^{(\iota)}$ as the ground set. The density decomposition with $\mathcal{I}^{(\iota)}$ as the ground set is a sequence of pairs $\{(S_\ell^{(0)}, S_\ell^{(1)})\}_{\ell \geq 1}$, together with a density vector $\rho_*^{(\iota)} : \mathcal{I}^{(\iota)} \to \mathbb{R}_+$, defined by the following iterative process.*

Initially, we set $G_0 := (\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$ to be the original given instance, and initialize $\ell = 1$.

1. *If $G_{\ell-1}$ is an instance where the vertex sets on both sides are empty, the process stops.*
2. *Otherwise, let $S_\ell^{(\iota)}$ be the maximal densest subset in $G_{\ell-1}$ with side ι being the ground set, where ρ_ℓ is the corresponding density; let $S_\ell^{(\bar{\iota})} := \mathcal{F}[S_\ell^{(\iota)}]$ be its closed neighborhood in $G_{\ell-1}$.*
   For each $u \in S_\ell^{(\iota)}$, define $\rho_*^{(\iota)}(u) := \rho_\ell = \frac{w(S_\ell^{(\bar{\iota})})}{w(S_\ell^{(\iota)})}$.
3. *Let G_ℓ be the sub-instance induced by removing $S_\ell^{(\iota)}$ and $S_\ell^{(\bar{\iota})}$ from $G_{\ell-1}$; note that edges in $\mathcal{F}$ incident to removed vertices are also removed.*
   Increment ℓ and go to the next iteration.

▶ Remark 17 (Isolated Vertices). Observe that if each side ι has isolated vertices $U^{(\iota)}$ that have no neighbors, then they will also appear at the beginning and the end of the decomposition sequence as $(U^{(0)}, \emptyset)$ and $(\emptyset, U^{(1)})$.

When $\mathcal{I}^{(\iota)}$ is the ground set, the density $\rho^{(\iota)}$ of vertices in $U^{(\iota)}$ is zero. In the rest of this section, we may assume that isolated vertices are removed from the instance.

The following fact has been proved in [15] when the vertices in the ground set have uniform weights. The same proof generalizes to the case where both sides have general positive weights.

▶ **Fact 18** (Local Maximin Condition Gives Density Decomposition). *Given an instance $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$, consider side $\iota \in \mathbb{B}$ as the ground set. Suppose $\alpha^{(\bar{\iota})} \in \mathbb{R}^{\mathcal{F}}$ is a refinement of the vertex weights $w^{(\bar{\iota})}$ on side $\bar{\iota}$, which induces the payload density vector $\rho^{(\iota)}$ on side ι as in Definition 8.*

Suppose further that $\alpha^{(\bar{\iota})}$ satisfies the local maximin condition in Definition 10. Then, $\rho^{(\iota)}$ coincides exactly with the density vector $\rho_*^{(\iota)}$ achieved by the density decomposition in Definition 16 with side ι being the ground set.

Moreover, if $(S_\ell^{(\bar{\iota})}, S_\ell^{(\iota)})$ is a pair in the density decomposition, then $S_\ell^{(\bar{\iota})}$ is exactly the collection of vertices that send positive payload to $S_\ell^{(\iota)}$ in $\alpha^{(\bar{\iota})}$.

Symmetry in Density Decomposition. In Definition 16, one side $\iota \in \mathbb{B}$ plays the role of the ground set. If we switch the ground set to the other side $\bar{\iota}$, the same sequence $\{(S_\ell^{(0)}, S_\ell^{(1)})\}_{\ell \geq 1}$ will be produced in the density decomposition, but in the reversed order. While a similar observation has been made by Harb et al. [23, Theorem 13], we re-establish this fact by considering the proportional response to a locally maximin refinement in the following lemma, whose proof is given in the full version [12].

▶ **Lemma 19** (Proportional Response to Locally Maximin Refinement). *Let $\alpha^{(0)}$ be a refinement of vertex weights in $\mathcal{I}^{(0)}$ that satisfies the local maximin condition. By Remark 11, suppose the refinement $\alpha^{(1)}$ on vertex weights in $\mathcal{I}^{(1)}$ is the unique proportional response to $\alpha^{(0)}$. Moreover, for each side $\iota \in \mathbb{B}$, the refinement $\alpha^{(\iota)}$ induces the payload density vector $\rho^{(\bar{\iota})}$ on vertices on the other side $\bar{\iota}$. Then, the following holds:*

- *For any $i \in \mathcal{I}^{(0)}$ and $j \in \mathcal{I}^{(1)}$ such that $i \sim j$ and $\alpha^{(0)}(ij) > 0$, their received payload densities due to the refinements satisfy $\rho^{(0)}(i) = \frac{1}{\rho^{(1)}(j)}$.*
- *The refinement $\alpha^{(1)}$ also satisfies the local maximin condition.*
- *We get back $\alpha^{(0)}$ as the proportional response to $\alpha^{(1)}$.*

In addition to being used to establish the symmetry of the density decomposition in Theorem 20, Lemma 19 will be used later as a technical tool for analyzing the universal refinement matching problem in Section 4.

▶ **Theorem 20** (Folklore: Symmetry of Density Decomposition). *Given an instance $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$, suppose the sequence $\{(S_\ell^{(0)}, S_\ell^{(1)})\}_{\ell \geq 1}$ of pairs is the density decomposition using $\mathcal{I}^{(1)}$ as the ground set.*

Then, the density decomposition using $\mathcal{I}^{(0)}$ as the ground set produces the same sequence of pairs in reversed order.

4 Universal Refinement Matching Problem

As mentioned in Section 1, an instance in Definition 6 can be interpreted as inputs in various problems. As a precursor to the universally closest distribution refinements problem studied in Section 5, we consider a matching problem. Here, an instance is interpreted as the input of the vertex-constrained fractional matching problem, where the vertex constraints on each side is scaled by the same multiplicative factor.

▶ **Definition 21** (Vertex-Constrained Matching Instance). *Given an instance $G = (\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$ as in Definition 6 and two non-negative weights $\vec{c} = (c^{(0)}, c^{(1)}) \in \mathbb{R}^{\mathbb{B}}$, we use $G^{(\vec{c})}$ to denote the instance of (fractional) matching on the bipartite graph $(\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F})$ with vertex capacity constraints. For side ι and $i \in \mathcal{I}^{(\iota)}$, the capacity constraint at vertex i is $c^{(\iota)} \cdot w(i)$.*

LP Interpretation. The maximum (fractional) matching problem instance $G^{(\vec{c})}$ is captured by the following linear program.

$$\begin{aligned} \mathsf{LP}(G^{(\vec{c})}) \qquad \max \quad & \sum_{i \sim j} x_{ij} \\ \text{s.t.} \quad & \sum_{j' : i \sim j'} x_{ij'} \leq c^{(0)} \cdot w(i), \quad \forall i \in \mathcal{I}^{(0)} \\ & \sum_{i' : i' \sim j} x_{i'j} \leq c^{(1)} \cdot w(j), \quad \forall j \in \mathcal{I}^{(1)} \\ & x_{ij} \geq 0, \quad \forall i \in \mathcal{I}^{(0)} \sim j \in \mathcal{I}^{(1)} \end{aligned}$$

We will also consider the dual in our analysis:

$$\begin{aligned} \mathsf{DP}(G^{(\vec{c})}) \qquad \min \quad & c^{(0)} \sum_{i \in \mathcal{I}^{(0)}} w(i) \cdot \lambda_i + c^{(1)} \sum_{j \in \mathcal{I}^{(1)}} w(j) \cdot \lambda_j \\ \text{s.t.} \quad & \lambda_i + \lambda_j \geq 1, \quad \forall i \in \mathcal{I}^{(0)} \sim j \in \mathcal{I}^{(1)} \\ & \lambda_i \geq 0, \quad \forall i \in \mathcal{I}^{(0)} \cup \mathcal{I}^{(1)} \end{aligned}$$

▶ **Definition 22** (Fractional Matching Induced by Refinement Pair). *Given an instance* $G = (\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$ *and a non-negative weight pair* $\vec{c} = (c^{(0)}, c^{(1)})$*, a refinement pair* $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$*, where* $\alpha^{(\iota)}$ *is a refinement on vertex weights in* $\mathcal{I}^{(\iota)}$*, induces a feasible fractional matching for* $G^{(\vec{c})}$ *that is denoted as follows:*

$\vec{c} \bullet \vec{\alpha}(ij) := \min\{c^{(0)} \cdot \alpha^{(0)}(ij), c^{(1)} \cdot \alpha^{(1)}(ij)\}$*, for all* $i \sim j$*.*

▶ Remark 23. The feasibility of the fractional matching $\vec{c} \bullet \vec{\alpha}$ to the instance $G^{(\vec{c})}$ follows directly because each $\alpha^{(\iota)}$ is a refinement of the vertex weights in $\mathcal{I}^{(\iota)}$.

▶ **Definition 24** (Universal Refinement Pair for Matching). *Given an instance* $G = (\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$ *and* $\tau \in [0, 1]$*, a refinement pair* $\vec{\alpha}$ *is* $(1-\tau)$*-universal for matching in* G*, if for any non-negative weight pair* $\vec{c}$*, the solution* $\vec{c} \bullet \vec{\alpha}$ *achieves* $(1-\tau)$*-approximation for the maximum matching problem on* $G^{(\vec{c})}$*.*

For $\tau = 0$*, we simply say* $\vec{\alpha}$ *is universal (for matching in* G*).*

Intuition for Universal Refinement Pair. Besides being a technical tool for later sections, the notion of a universal refinement pair is interesting in terms of understanding the structure of vertex-constrained bipartite matching. Suppose it is known that the vertex weights on each side are correlated and may vary but follow some fixed relative ratios among themselves. The idea of a universal refinement pair is that some pre-computation can be performed such that when the vertex weights on each side change proportionally, the matching can be updated quickly to get a (nearly) optimal solution without computing from scratch.

Observe that the density decomposition in Definition 16 does not change if the vertex weights on one side are multiplied by the same scalar. Informally, the decomposition identifies different levels of "congestion" when a matching is performed between vertices on the two sides. The following result formalizes the connection between the density decomposition and universal refinement pairs.

▶ **Lemma 25** (Local Maximin Achieves Universal for Matching). *Suppose in an instance* G*, for side* $\iota \in \mathbb{B}$*, the refinement* $\alpha^{(\iota)}$ *on vertex weights in* $\mathcal{I}^{(\iota)}$ *is locally maximin, and the refinement* $\alpha^{(\bar{\iota})}$ *is the (unique) proportional response to* $\alpha^{(\iota)}$*. Then, the refinement pair* $(\alpha^{(\iota)}, \alpha^{(\bar{\iota})})$ *is universal for matching in* G*.*

Moreover, suppose in the density decomposition, for side ι*,* $\rho_*^{(\iota)}$ *is the payload density vector and* $p_*^{(\iota)}$ *is the payload vector. Then, for any* $c = (c^{(\iota)}, c^{(\bar{\iota})})$*, the weight of the maximum matching in* $G^{(\vec{c})}$ *is:*

$\sum_{j \in \mathcal{I}^{(\iota)}} \min\{\frac{c^{(\iota)}}{\rho_*^{(\iota)}(j)}, c^{(\bar{\iota})}\} \cdot p_*^{(\iota)}(j)$.

Proof. By Lemma 19, $\alpha^{(\iota)}$ and $\alpha^{(\bar{\iota})}$ are proportional responses to each other and both are locally maximin. Fix some $\vec{c} = (c^{(0)}, c^{(1)})$ and we show that $\vec{c} \bullet \vec{\alpha}$ is an optimal solution to $G^{(\vec{c})}$.

Let $\{(S_\ell^{(0)}, S_\ell^{(1)}\}_{\ell \geq 1}$ be the density decomposition with $\mathcal{I}^{(1)}$ being the ground set. (Recall that the density decomposition is symmetric between $\mathcal{I}^{(0)}$ and $\mathcal{I}^{(1)}$, where the forward or backward order of the sequence depends on which side is the ground set.)

From Fact 18, recall that $\alpha^{(0)}(ij)$ and $\alpha^{(1)}(ij)$ are non-zero only if there exists some ℓ such that $(i, j) \in \mathcal{F}_\ell := \mathcal{F} \cap (S_\ell^{(0)} \times S_\ell^{(1)})$.

Let $L^{(1)} := \{\ell : c^{(1)} \cdot w(S_\ell^{(1)}) \leq c^{(0)} \cdot w(S_\ell^{(0)})\}$ and $L^{(0)} := \{\ell : c^{(1)} \cdot w(S_\ell^{(1)}) > c^{(0)} \cdot w(S_\ell^{(0)})\}$.

Observe that for $i \in S_\ell^{(1)}$, $\rho_*^{(1)}(i) = \frac{w(S_\ell^{(0)})}{w(S_\ell^{(1)})}$. Therefore, for all $\ell \in L^{(1)}$ and $\ell' \in L^{(0)}$, $\ell < \ell'$.

Moreover, for all $\ell < \ell'$, for all $i \in S_\ell^{(0)}$ and $j \in S_{\ell'}^{(1)}$, i and j are not neighbors in $\mathcal{F}$, from the definition of the density decomposition.

Hence, we can construct a feasible dual solution λ to $\mathsf{DP}(G^{(\vec{c})})$ as follows.

- For $\ell \in L^{(1)}$: for $i \in S_\ell^{(0)}$, $\lambda_i = 0$; for $j \in S_\ell^{(1)}$, $\lambda_j = 1$.
- For $\ell \in L^{(0)}$: for $i \in S_\ell^{(0)}$, $\lambda_i = 1$; for $j \in S_\ell^{(1)}$, $\lambda_j = 0$.

The dual feasibility of λ holds because if $i \in \mathcal{I}^{(0)}$ and $j \in \mathcal{I}^{(1)}$ such that $\lambda_i = \lambda_j = 0$, then it must be the case that i and j are not neighbors in $\mathcal{F}$.

Next, we check that the dual solution λ has the same objective value as the primal solution $\vec{c} \bullet \vec{\alpha}$ to $\mathsf{LP}(G^{\vec{c}})$.

Observe that for $\ell \in L^{(1)}$, for $i \in S_\ell^{(0)} \sim j \in S_\ell^{(1)}$, we have

$c^{(1)} \cdot \alpha^{(1)}(ij) = c^{(1)} \cdot \alpha^{(0)}(ij) \cdot \frac{w(S_\ell^{(1)})}{w(S_\ell^{(0)})} \leq c^{(0)} \cdot \alpha^{(0)}(ij)$.

Hence, $c^{(1)} \sum_{j \in S_\ell^{(1)}} w(j) \cdot \lambda_j = c^{(1)} \sum_{(i,j) \in \mathcal{F}_\ell} \alpha^{(1)}(ij) = \sum_{(i,j) \in \mathcal{F}_\ell} \vec{c} \bullet \vec{\alpha}(ij)$, where the first equality holds because $\alpha^{(1)}$ restricted to $\mathcal{F}_\ell$ is a refinement of vertex weights in $S_\ell^{(1)}$.

Similarly, for $\ell \in L^{(0)}$, we have $c^{(0)} \sum_{i \in S_\ell^{(0)}} w(i) \cdot \lambda_i = \sum_{(i,j) \in \mathcal{F}_\ell} \vec{c} \bullet \vec{\alpha}(ij)$.

Therefore, the feasible primal solution $\vec{c} \bullet \vec{\alpha}$ and the feasible dual solution λ have the same objective value, which implies that both are optimal.

Finally, we compute the weight of the maximum matching in $G^{(\vec{c})}$. For simpler notation, it is understood that in each sum below, we include only pairs (i,j) such that both $\alpha^{(0)}(ij)$ and $\alpha^{(1)}(ij)$ are non-zero.

$$
\begin{aligned}
&\sum_{(i,j) \in (\mathcal{I}^{(\bar{\iota})} \times \mathcal{I}^{(\iota)}) \cap \mathcal{F} : \alpha(ij) > 0} \min\{c^{(\bar{\iota})} \cdot \alpha^{(\bar{\iota})}(ij), c^{(\iota)} \cdot \alpha^{(\iota)}(ij)\} \\
&= \sum_{(i,j)} \min\{c^{(\bar{\iota})}, c^{(\iota)} \cdot \frac{\alpha^{(\iota)}(ij)}{\alpha^{(\bar{\iota})}(ij)}\} \cdot \alpha^{(\bar{\iota})}(ij) \\
&= \sum_{(i,j)} \min\{c^{(\bar{\iota})}, c^{(\iota)} \cdot \frac{1}{\rho_*^{(\iota)}(j)}\} \cdot \alpha^{(\bar{\iota})}(ij) \\
&= \sum_{j \in \mathcal{I}^{(\iota)}} \min\{c^{(\bar{\iota})}, \frac{c^{(\iota)}}{\rho_*^{(\iota)}(j)}\} \sum_{i : i \sim j} \alpha^{(\bar{\iota})}(ij) \\
&= \sum_{j \in \mathcal{I}^{(\iota)}} \min\{\frac{c^{(\iota)}}{\rho_*^{(\iota)}(j)}, c^{(\bar{\iota})}\} \cdot p_*^{(\iota)}(j).
\end{aligned}
$$

◀

5 Universally Closest Distribution Refinements Problem

In this section, we show that when an instance is interpreted as two distributions on two sample spaces imposed with a neighboring relation, a refinement pair satisfying Goal 3 is an optimal solution to the *universally closest distribution refinements problem*.

Distribution Instance. In Definition 6, an input is a *distribution instance*, if, in addition, we require that for each side $\iota \in \mathbb{B}$, the vertex weights satisfies $\sum_{i \in \mathcal{I}^{(\iota)}} w^{(\iota)}(i) = 1$, i.e., $w^{(\iota)}$ is a probability distribution on $\mathcal{I}^{(\iota)}$.

Observe that given a distribution instance $G = (\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$, for each side $\iota \in \mathbb{B}$, a refinement $\alpha^{(\iota)}$ of the vertex weights $w^{(\iota)}$ is a probability distribution on $\mathcal{F}$. A high-level problem is to find a refinement pair $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$ such that the distributions $\alpha^{(0)}$ and $\alpha^{(1)}$ on $\mathcal{F}$ are as "close" to each other as possible.

Divergence Notion. Given two distributions P and Q on the same sample space Ω, a notion of *divergence* $\mathsf{D}(P\|Q)$ measures how close two distributions are, where a smaller value typically means that the two distributions are closer. An example is the total variation distance $\mathsf{D}_{\mathrm{TV}}(P\|Q) := \sup_{E\subseteq\Omega} |P(E) - Q(E)|$. However, note that in general, a divergence needs not be symmetric, i.e., it is possible that $\mathsf{D}(P\|Q) \neq \mathsf{D}(Q\|P)$. An example of a non-symmetric divergence notion is $\mathsf{D}_{\mathrm{KL}}(P\|Q) = \sum_{\omega\in\Omega} p_\omega \ln \frac{p_\omega}{q_\omega}$.

▶ **Definition 26** (Closest Refinement Pair). *Given a distribution instance $G = (\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$ and a divergence notion D, a refinement pair $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$ (as in Definition 8) is closest (with respect to D) if, the pair attains the minimum $\mathsf{D}(\alpha^{(0)}||\alpha^{(1)})$ among all refinement pairs, whose value we denote as $\mathsf{D}^*(G)$.*

▶ Remark 27. Since a divergence is typically continuous and the collection of refinement pairs forms a compact set, the minimum is attainable in such a case.

However, note that since D is not necessarily symmetric, from Definition 26 alone, it is not clear if a pair minimizing $\mathsf{D}(\alpha^{(0)}||\alpha^{(1)})$ would also minimize $\mathsf{D}(\alpha^{(1)}||\alpha^{(0)})$.

▶ **Definition 28** (Universally Closest Refinement Pair). *Given a distribution instance $G = (\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$ and a collection $\mathcal{D}$ of divergence notions, a refinement pair $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$ is universally closest (with respect to $\mathcal{D}$) if, for all $\mathsf{D} \in \mathcal{D}$, $\mathsf{D}(\alpha^{(0)}||\alpha^{(1)}) = \mathsf{D}^*(G)$.*

5.1 Class of Hockey-Stick Divergences

We first consider the class of hockey-stick divergences.

▶ **Definition 29** (Hockey-Stick Divergence). *Given distributions P and Q on the same sample space Ω and $\gamma \geq 0$, the (reversed[6]) hockey-stick divergence is defined as*

$\mathsf{D}_\gamma(P\|Q) := \sup_{S\subseteq\Omega} Q(S) - \gamma \cdot P(S)$.

▶ Remark 30. The hockey-stick divergence is related to the well-known (ϵ, δ)-differential privacy inequality. Specifically, $\mathsf{D}_{e^\epsilon}(P\|Q) \leq \delta$ *iff* for all subsets $S \subseteq \Omega$,

$Q(S) \leq e^\epsilon \cdot P(S) + \delta$.

To express the other direction of the inequality, i.e., $\mathsf{D}_{e^\epsilon}(Q\|P) \leq \delta$, we could consider another parameter $\widehat{\gamma} \geq 0$ and keep the order of arguments in $\mathsf{D}_{\widehat{\gamma}}(P\|Q)$. Specifically, by considering the complement of events S in the above inequality, we can deduce that: $\mathsf{D}_{e^\epsilon}(Q\|P) \leq \delta$ *iff* $\mathsf{D}_{e^{-\epsilon}}(P\|Q) \leq 1 - e^{-\epsilon}(1-\delta)$.

We consider universal closest distribution refinements with respect to the class of hockey-stick divergences $\mathcal{D}_{\mathrm{HS}} := \{\mathsf{D}_\gamma : \gamma \geq 0\}$.

The following lemma shows that the hockey-stick divergence is closely related to the matching instance defined in Definition 21.

▶ **Lemma 31** (Hockey-Stick Divergence and Matching). *Suppose $G = (\mathcal{I}^{(0)}, \mathcal{I}^{(1)}; \mathcal{F}; w)$ is a distribution instance and $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$ is a refinement pair. For $\gamma \geq 0$, consider the pair of weights $\vec{c} := (c^{(0)} = \gamma, c^{(1)} = 1)$.*

Then, the weight of the fractional matching $\vec{c} \bullet \vec{\alpha}$ in the matching instance $G^{(\vec{c})}$ is exactly $1 - \mathsf{D}_\gamma(\alpha^{(0)}\|\alpha^{(1)})$.

In order words, finding a closest refinement pair for the divergence D_γ is equivalent to finding a maximum matching in $G^{(\vec{c})}$.

[6] In the literature, the hockey-stick divergence $\widetilde{\mathsf{D}}_\gamma$ is defined with the roles of P and Q interchanged: $\widetilde{\mathsf{D}}_\gamma(P\|Q) = \mathsf{D}_\gamma(Q\|P)$. We change the convention to make the notation consistent when we consider power functions later.

Proof. Let $S := \{f \in \mathcal{F} : \alpha^{(1)}(f) \geq \gamma \cdot \alpha^{(0)}(f)\}$.
Observe that $\mathsf{D}_\gamma(\alpha^{(0)} \| \alpha^{(1)}) = \sum_{f \in S}(\alpha^{(1)}(f) - \gamma \cdot \alpha^{(0)}(f))$.
Finally, the weight of the matching $\vec{c} \bullet \vec{\alpha}$ is:

$$\sum_{f \in \mathcal{F}} \min\{\gamma \cdot \alpha^{(0)}(f), \alpha^{(1)}(f)\} = \sum_{f \in S} \gamma \cdot \alpha^{(0)}(f) + \sum_{f \in \mathcal{F} \setminus S} \alpha^{(1)}(f)$$
$$= 1 - \sum_{f \in S}(\alpha^{(1)}(f) - \gamma \cdot \alpha^{(0)}(f)) = 1 - \mathsf{D}_\gamma(\alpha^{(0)} \| \alpha^{(1)}),$$

as required. ◀

▶ **Theorem 32** (Locally Maximin Pair is Closest for $\mathcal{D}_{\mathrm{HS}}$)**.** *Suppose in a distribution instance G, $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$ is a distribution refinement pair that is locally maximin and proportional response to each other. Then, $\vec{\alpha}$ is universally closest with respect to the class $\mathcal{D}_{\mathrm{HS}}$ of hockey-stick divergences.*

Proof. The statement is a direct consequence of Lemmas 25 and 31. ◀

5.2 Class of Data Processing Divergences

Extending Theorem 32, we show that a locally maximin pair is actually the closest with respect to a wide class of divergences. In fact, for any reasonable usage of divergence, it is intuitive that taking a partial view of a pair of objects should not increase the divergence between them. This is formally captured by the *data processing inequality*.

▶ **Definition 33** (Data Processing Inequality)**.** *A divergence notion* D *satisfies the data processing inequality if given any two distributions P and Q on the sample space Ω and any function $\varphi : \Omega \to \widehat{\Omega}$, the induced distributions $\varphi(P)$ and $\varphi(Q)$ on $\widehat{\Omega}$ satisfy the monotone property:*

$$\mathsf{D}(\varphi(P) \| \varphi(Q)) \leq \mathsf{D}(P \| Q).$$

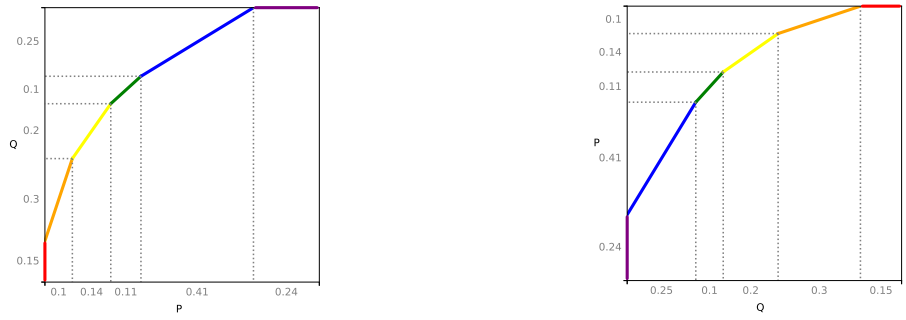
We use $\mathcal{D}_{\mathrm{DPI}}$ to denote the class of divergences satisfying the data processing inequality.

It is known that all divergences in $\mathcal{D}_{\mathrm{DPI}}$ between two distributions P and Q can be recovered from the power function between them. Recall that in this work, we focus on finite sample spaces; see Figure 1 for an example. It is known that power functions have a fractional knapsack interpretation [25].

▶ **Definition 34** (Power Function)**.** *Suppose P and Q are distributions on the same sample space Ω.*

- *For a discrete sample space Ω, the power function* $\mathsf{Pow}(P \| Q) : [0,1] \to [0,1]$ *can be defined in terms of the* fractional knapsack problem.
  Given a collection Ω of items, suppose $\omega \in \Omega$ has weight $P(\omega)$ and value $Q(\omega)$. Then, given $x \in [0,1]$, $\mathsf{Pow}(P \| Q)(x)$ is the maximum value attained with weight capacity constraint x, where items may be taken fractionally.
- *For a continuous sample space Ω, given $x \in [0,1]$, $\mathsf{Pow}(P \| Q)(x)$ is the supremum of $Q(S)$ over measurable subsets $S \subseteq \Omega$ satisfying $P(S) = x$.*

In the literature (see [16] for instance), the tradeoff function $\mathsf{T}(P \| Q) := 1 - \mathsf{Pow}(P \| Q)$, which has essentially equivalent mathematical properties, has also been considered. However, power functions have the same notation interpretation as divergences in the sense that



(a) $g = \mathsf{Pow}(P||Q)$. **(b)** $\widehat{g} = \mathsf{Pow}(Q\|P) = \mathfrak{R}(\mathsf{Pow}(P\|Q))$.

Figure 1 Consider a sample space Ω with 6 elements indicated with different rainbow colors. Two distributions on Ω are given by the arrays: $P = [0, 0.1, 0.14, 0.11, 0.41, 0.24]$ and $Q = [0.15, 0.3, 0.2, 0.1, 0.25, 0]$, where the elements $\omega \in \Omega$ are sorted in increasing order of the ratio $\frac{Q(\omega)}{P(\omega)}$. The left figure shows the power curve $g = \mathsf{Pow}(P||Q)$, and the right figure shows $\widehat{g} = \mathsf{Pow}(Q\|P)$ which is the reflection of g about the line $y = 1 - x$.

"smaller" means "closer" distributions. In fact, Vadhan and Zhang [38] defined an abstract concept of *generalized probability distance* (with the tradeoff function as the main example) such that the notation agrees with this interpretation. We recap some well-known properties of power functions.

▶ **Fact 35** (Valid power functions [16, Proposition 2.2]). *Suppose $g : [0,1] \to [0,1]$ is a function. Then, there exist distributions X and Y such that $\mathsf{Pow}(X\|Y) = g$ iff g is continuous, concave and $g(x) \geq x$ for all $x \in [0,1]$.*

Partial Order on Power Functions. We consider a partial order on power functions. Given such functions g_1 and g_2, we denote $g_1 \preceq g_2$ *iff* for all $x \in [0,1]$, $g_1(x) \leq g_2(x)$.

We next state formally how a divergence satisfying Definition 33 can be recovered from the power function.

▶ **Fact 36** (Recovering Divergence From Power Function [16, Proposition B.1]). *For any divergence $\mathsf{D} \in \mathcal{D}_{\mathrm{DPI}}$, there exists a functional $\ell_{\mathsf{D}} : [0,1]^{[0,1]} \to \mathbb{R}$ such that for any distributions P and Q on the same sample space, $\mathsf{D}(P\|Q) = \ell_{\mathsf{D}}(\mathsf{Pow}(P\|Q))$.*

Moreover, if $g_1 \preceq g_2$, then $\ell_{\mathsf{D}}(g_1) \leq \ell_{\mathsf{D}}(g_2)$.

The following is the main result of this section, which states that a locally maximin pair essentially solves the closest distribution refinement problem optimally in the most general sense.

▶ **Theorem 37** (Locally Maximin Pair Gives Minimal Power Function). *Suppose in a distribution instance G, $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)})$ is a distribution refinement pair that is locally maximin and proportional response to each other. Then, the power function $g_\dagger$ between any other distribution refinement pair of G satisfies $\mathsf{Pow}(\alpha^{(0)}\|\alpha^{(1)}) \preceq g_\dagger$. Consequently, any such locally maximin pair gives rise to the same power function, and we denote $\mathsf{Pow}^*(G) = \mathsf{Pow}(\alpha^{(0)}\|\alpha^{(1)})$.*

From Fact 36, this implies that $\vec{\alpha}$ is a universally closest refinement pair with respect to $\mathcal{D}_{\mathrm{DPI}}$.

▶ Remark 38. Observe that given any $\mathsf{D} \in \mathcal{D}_{\mathrm{DPI}}$ and any strictly increasing function $\phi : \mathbb{R} \to \mathbb{R}$, we still have $\phi(\mathsf{D}(\cdot\|\cdot)) \in \mathcal{D}_{\mathrm{DPI}}$. Since any deviation from the correct answer can be arbitrarily magnified by ϕ, it is difficult to give any meaningful approximation guarantee on the value of a general divergence in $\mathcal{D}_{\mathrm{DPI}}$.

We prove Theorem 37 by analyzing the relationship between hockey-stick divergence and power function, and extend the result from Theorem 32. Similar to convex functions, we also consider the notion of subgradient for power functions.

▶ **Definition 39** (Subgradient). *Suppose $g : [0,1] \to [0,1]$ is a power function and $x \in [0,1]$. Then, the subgradient of g at x is defined as the collection $\partial g(x)$ of real numbers satisfying:*

$\gamma \in \partial g(x)$ iff *for all* $y \in [0,1]$, $g(x) + \gamma \cdot (y - x) \geq g(y)$.

In other words, the line segment with slope γ touching the curve g at x never goes below the curve.

The following is a special case of Fact 36, and describes explicitly how a hockey-stick divergence is recovered from a power function. See Figure 2 for an illustration.

▶ **Fact 40** (Recovering Hockey-Stick Divergence From Power Function). *Suppose P and Q are distributions on the same sample space Ω, where $g = \mathsf{Pow}(P\|Q)$ is the power function between them. Then, for any $\gamma \geq 0$, there exists $x \in [0,1]$ such that $\gamma \in \partial g(x)$ and $\mathsf{D}_\gamma(P\|Q) = g(x) - \gamma x$.*

In other words, one can find a line segment with slope γ touching the curve g and the y-intercept of the line will give $\mathsf{D}_\gamma(P\|Q)$.

Proof. The result should be standard, but we include a short proof for completeness. One can imagine a line segment with slope γ starting high above and gradually being lowered until touching the curve g at some $x \in [0,1]$. Observe that the line segment may touch the curve at more than 1 point, in which case we take x to be the infimum.

We partition the sample space Ω into $S_1 := \{\omega \in \Omega : Q(\omega) > \gamma \cdot P(\omega)\}$ and $S_2 := \Omega \setminus S_1$. Observe that $P(S_1) = x$ and $g(x) = Q(S_1)$.

Then, it follows that $\mathsf{D}_\gamma(P\|Q) = Q(S_1) - \gamma \cdot P(S_1) = g(x) - \gamma x$, as required. ◀

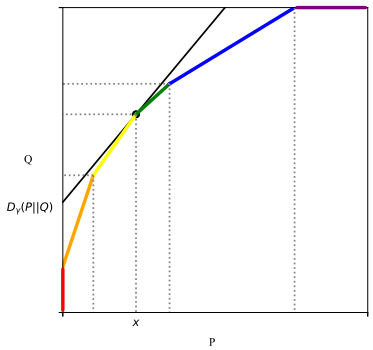


Figure 2 Using the same example in Figure 1, consider a line segment with slope $\gamma = 1.2$ touching the curve g at x. In this case, the red, orange, and yellow elements ω to the left of point x satisfy $\frac{Q(\omega)}{P(\omega)} > 1.2$.

Proof of Theorem 37. Suppose $g = \mathsf{Pow}(\alpha^{(0)}\|\alpha^{(1)})$ is the power function between the given locally maximin pair $\vec{\alpha}$. For the sake of contradiction, suppose some other refinement pair $\vec{\alpha}_\dagger$ produces the power function $g_\dagger$ such that for some $x \in [0,1]$, $g(x) > g_\dagger(x)$.

Fix some $\gamma \in \partial g_\dagger(x)$ and consider the line segment $\ell_\dagger$ with slope γ touching curve $g_\dagger$ at x. By Fact 40, the hockey-stick divergence $\mathsf{D}_\gamma(\vec{\alpha}_\dagger)$ is the y-intercept of $\ell_\dagger$.

However, since $g(x) > g_\dagger(x)$, to find a line segment with slope γ touching curve g, we must move $\ell_\dagger$ strictly upwards. This implies that $\mathsf{D}_\gamma(\vec{\alpha}) > \mathsf{D}_\gamma(\vec{\alpha}_\dagger)$, contradicting Theorem 32 which states that $\vec{\alpha}$ is universally closest with respect to $\mathcal{D}_{\text{HS}}$. ◀

References

1 Albert Angel, Nikos Sarkas, Nick Koudas, and Divesh Srivastava. Dense subgraph maintenance under streaming edge weight updates for real-time story identification. *Proceedings of the VLDB Endowment*, 5(6):574–585, 2012. doi:10.14778/2168651.2168658.

2 Kenneth J Arrow and Gerard Debreu. Existence of an equilibrium for a competitive economy. *Econometrica: Journal of the Econometric Society*, pages 265–290, 1954.

3 Oana Denisa Balalau, Francesco Bonchi, T.-H. Hubert Chan, Francesco Gullo, and Mauro Sozio. Finding subgraphs with maximum total density and limited overlap. In *WSDM*, pages 379–388. ACM, 2015. doi:10.1145/2684822.2685298.

4 Nikhil Bansal and Ilan Reuven Cohen. Contention resolution, matrix scaling and fair allocation. In *WAOA*, volume 12982 of *Lecture Notes in Computer Science*, pages 252–274. Springer, 2021. doi:10.1007/978-3-030-92702-8_16.

5 Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009. doi:10.1137/080716542.

6 Roger E Behrend. Fractional perfect b-matching polytopes i: General theory. *Linear Algebra and its Applications*, 439(12):3822–3858, 2013.

7 Aaron Bernstein, Jacob Holm, and Eva Rotenberg. Online bipartite matching with amortized $O(\log^2 n)$ replacements. *J. ACM*, 66(5):37:1–37:23, 2019. doi:10.1145/3344999.

8 Benjamin Birnbaum, Nikhil R Devanur, and Lin Xiao. Distributed algorithms via gradient descent for fisher markets. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 127–136, 2011. doi:10.1145/1993574.1993594.

9 Richard A Brualdi. *Combinatorial matrix classes*, volume 13. Cambridge University Press, 2006.

10 Niv Buchbinder, Anupam Gupta, Daniel Hathcock, Anna R. Karlin, and Sherry Sarkar. Maintaining matroid intersections online. In *SODA*, pages 4283–4304. SIAM, 2024. doi:10.1137/1.9781611977912.149.

11 T.-H. Hubert Chan, Anand Louis, Zhihao Gavin Tang, and Chenzi Zhang. Spectral properties of hypergraph laplacian and approximation algorithms. *J. ACM*, 65(3):15:1–15:48, 2018. doi:10.1145/3178123.

12 T.-H. Hubert Chan and Quan Xue. Symmetric splendor: Unraveling universally closest refinements and fisher market equilibrium through density-friendly decomposition. *CoRR*, abs/2406.17964, 2024. doi:10.48550/arXiv.2406.17964.

13 Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In Klaus Jansen and Samir Khuller, editors, *Approximation Algorithms for Combinatorial Optimization*, pages 84–95, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. doi:10.1007/3-540-44436-X_10.

14 Li Chen, Rasmus Kyng, Yang P. Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Maximum flow and minimum-cost flow in almost-linear time. In *FOCS*, pages 612–623. IEEE, 2022. doi:10.1109/FOCS54457.2022.00064.

15 Maximilien Danisch, T.-H. Hubert Chan, and Mauro Sozio. Large scale density-friendly graph decomposition via convex programming. In *WWW*, pages 233–242. ACM, 2017. doi:10.1145/3038912.3052619.

16 Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):3–37, 2022.

17 Cynthia Dwork. Differential privacy. In *ICALP (2)*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006. doi:10.1007/11787006_1.

18 Irving Fisher. Mathematical investigations in the theory of value and prices, 1892.

19 Satoru Fujishige. Lexicographically optimal base of a polymatroid with respect to a weight vector. *Mathematics of Operations Research*, 5(2):186–196, 1980. doi:10.1287/MOOR.5.2.186.

20 Ashish Goel, Michael Kapralov, and Sanjeev Khanna. On the communication and streaming complexity of maximum bipartite matching. In *SODA*, pages 468–485. SIAM, 2012. doi:10.1137/1.9781611973099.41.

21 Andrew V Goldberg. Finding a maximum density subgraph, 1984.

22 Elfarouk Harb, Kent Quanrud, and Chandra Chekuri. Faster and scalable algorithms for densest subgraph and decomposition. *Advances in Neural Information Processing Systems*, 35:26966–26979, 2022.

23 Elfarouk Harb, Kent Quanrud, and Chandra Chekuri. Convergence to lexicographically optimal base in a (contra)polymatroid and applications to densest subgraph and tree packing. In *ESA*, volume 274 of *LIPIcs*, pages 56:1–56:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.ESA.2023.56.

24 Kamal Jain and Vijay V. Vazirani. Eisenberg-gale markets: Algorithms and game-theoretic properties. *Games Econ. Behav.*, 70(1):84–106, 2010. doi:10.1016/J.GEB.2008.11.011.

25 Joseph B Kadane. Discrete search and the neyman-pearson lemma. *Journal of Mathematical Analysis and Applications*, 22(1):156–171, 1968.

26 Samir Khuller and Barna Saha. On finding dense subgraphs. In *International colloquium on automata, languages, and programming*, pages 597–608. Springer, 2009. doi:10.1007/978-3-642-02927-1_50.

27 Bernhard H. Korte. *Combinatorial optimization: theory and algorithms*. Springer, 2000.

28 Euiwoong Lee and Sahil Singla. Maximum matching in the online batch-arrival model. *ACM Trans. Algorithms*, 16(4):49:1–49:31, 2020. doi:10.1145/3399676.

29 Xiangfeng Li, Shenghua Liu, Zifeng Li, Xiaotian Han, Chuan Shi, Bryan Hooi, He Huang, and Xueqi Cheng. Flowscope: Spotting money laundering based on graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34(04), pages 4731–4738, 2020. doi:10.1609/AAAI.V34I04.5906.

30 Chenhao Ma, Yixiang Fang, Reynold Cheng, Laks VS Lakshmanan, Wenjie Zhang, and Xuemin Lin. Efficient algorithms for densest subgraph discovery on large directed graphs. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 1051–1066, 2020. doi:10.1145/3318464.3389697.

31 Ilya Mironov. Rényi differential privacy. In *CSF*, pages 263–275. IEEE Computer Society, 2017. doi:10.1109/CSF.2017.11.

32 Rajeev Motwani, Rina Panigrahy, and Ying Xu. Fractional matching via balls-and-bins. In *APPROX-RANDOM*, volume 4110 of *Lecture Notes in Computer Science*, pages 487–498. Springer, 2006. doi:10.1007/11830924_44.

33 Yurii Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, 269:543–547, 1983. URL: https://api.semanticscholar.org/CorpusID:145918791.

34 Alexander Schrijver et al. *Combinatorial optimization: polyhedra and efficiency*, volume 24(2). Springer, 2003.

35 Kijung Shin, Tina Eliassi-Rad, and Christos Faloutsos. Corescope: Graph mining using k-core analysis – Patterns, anomalies and algorithms. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 469–478. IEEE, 2016. doi:10.1109/ICDM.2016.0058.

36 Nikolaj Tatti. Density-friendly graph decomposition. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(5):1–29, 2019. doi:10.1145/3344210.

37 Nikolaj Tatti and Aristides Gionis. Density-friendly graph decomposition. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1089–1099, 2015. doi:10.1145/2736277.2741119.

38 Salil P. Vadhan and Wanrong Zhang. Concurrent composition theorems for differential privacy. In *STOC*, pages 507–519. ACM, 2023. doi:10.1145/3564246.3585241.

39 Fang Wu and Li Zhang. Proportional response dynamics leads to market equilibrium. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 354–363, 2007. doi:10.1145/1250790.1250844.

40 Li Zhang. Proportional response dynamics in the fisher market. In *ICALP (2)*, volume 5556 of *Lecture Notes in Computer Science*, pages 583–594. Springer, 2009. doi:10.1007/978-3-642-02930-1_48.

41 Mingxun Zhou, Elaine Shi, T.-H. Hubert Chan, and Shir Maimon. A theory of composition for differential obliviousness. In *EUROCRYPT (3)*, volume 14006 of *Lecture Notes in Computer Science*, pages 3–34. Springer, 2023. doi:10.1007/978-3-031-30620-4_1.

42 Mingxun Zhou, Mengshi Zhao, T.-H. Hubert Chan, and Elaine Shi. Advanced composition theorems for differential obliviousness. In *ITCS*, volume 287 of *LIPIcs*, pages 103:1–103:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.ITCS.2024.103.

Entry-Specific Matrix Estimation Under Arbitrary Sampling Patterns Through the Lens of Network Flows

Yudong Chen
Department of Computer Sciences, University of Wisconsin-Madison, WI, USA

Xumei Xi
School of Operations Research and Information Engineering, Cornell University, Ithaca, NY, USA

Christina Lee Yu
School of Operations Research and Information Engineering, Cornell University, Ithaca, NY, USA

Abstract

Matrix completion tackles the task of predicting missing values in a low-rank matrix based on a sparse set of observed entries. It is often assumed that the observation pattern is generated uniformly at random or has a very specific structure tuned to a given algorithm. There is still a gap in our understanding when it comes to arbitrary sampling patterns. Given an arbitrary sampling pattern, we introduce a matrix completion algorithm based on network flows in the bipartite graph induced by the observation pattern. For additive matrices, we show that the electrical flow is optimal, and we establish error upper bounds customized to each entry as a function of the observation set, along with matching minimax lower bounds. Our results show that the minimax squared error for recovery of a particular entry in the matrix is proportional to the effective resistance of the corresponding edge in the graph. Furthermore, we show that the electrical flow estimator is equivalent to the least squares estimator. We apply our estimator to the two-way fixed effects model and show that it enables us to accurately infer individual causal effects and the unit-specific and time-specific confounders. For rank-1 matrices, we use edge-disjoint paths to form an estimator that achieves minimax optimal estimation when the sampling is sufficiently dense. Our discovery introduces a new family of estimators parametrized by network flows, which provide a fine-grained and intuitive understanding of the impact of the given sampling pattern on the difficulty of estimation at an entry-specific level. This graph-based approach allows us to quantify the inherent complexity of matrix completion for individual entries, rather than relying solely on global measures of performance.

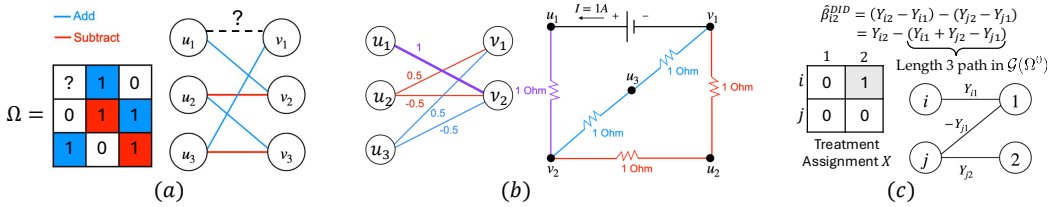


Figure 1 (a) Example of transforming a matrix observation pattern to a bipartite graph. Each a path from u_1 to v_1 defines an unbiased estimator for the ? entry. (b) Example of a network flow, in particular the electrical flow that corresponds to the optimal unbiased estimator. This flow sends a unit of current from u_1 to v_1 on the electrical network constructed by treating each edge in the graph as a unit resistor, thus putting high weight on edges that appear on multiple paths. (c) Equivalence of Difference-in-differences estimators to flow estimators constrained to length-3 paths.

2012 ACM Subject Classification Networks → Network algorithms

Keywords and phrases Matrix estimation, network flows, panel data analysis, non-uniform sampling

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.36

© Yudong Chen, Xumei Xi, and Christina Lee Yu;
licensed under Creative Commons License CC-BY 4.0
16th Innovations in Theoretical Computer Science Conference (ITCS 2025).
Editor: Raghu Meka; Article No. 36; pp. 36:1–36:2
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Category Extended Abstract

Related Version *Full Version*: https://www.arxiv.org/pdf/2409.03980

Funding National Science Foundation grants CCF-1704828, CCF-2233152, CCF-2337796 and CNS-1955997, and AFOSR grant FA9550-23-1-0301

A Lower Bound on the Trace Norm of Boolean Matrices and Its Applications

Tsun-Ming Cheung
School of Computer Science, McGill University, Montreal, Canada

Hamed Hatami
School of Computer Science, McGill University, Montreal, Canada

Kaave Hosseini
Department of Computer Science, University of Rochester, NY, USA

Aleksandar Nikolov
Department of Computer Science, University of Toronto, Canada

Toniann Pitassi
Department of Computer Science, Columbia University, New York, NY, USA

Morgan Shirley
Department of Computer Science, University of Victoria, Canada

Abstract

We present a simple method based on a variant of Hölder's inequality to lower-bound the trace norm of Boolean matrices. As the main result, we obtain an exponential separation between the randomized decision tree depth and the spectral norm (i.e. the Fourier L_1-norm) of a Boolean function. This answers an open question of Cheung, Hatami, Hosseini and Shirley (CCC 2023). As immediate consequences, we obtain the following results.

- We give an exponential separation between the logarithm of the randomized and the deterministic parity decision tree *size*. This is in sharp contrast with the standard binary decision tree setting where the logarithms of randomized and deterministic decision tree size are essentially polynomially related, as shown recently by Chattopadhyay, Dahiya, Mande, Radhakrishnan, and Sanyal (STOC 2023).
- We give an exponential separation between the approximate and the exact spectral norm for Boolean functions.
- We give an exponential separation for XOR functions between the deterministic communication complexity with oracle access to Equality function (D^{EQ}) and randomized communication complexity. Previously, such a separation was known for general Boolean matrices by Chattopadhyay, Lovett, and Vinyals (CCC 2019) using the Integer Inner Product (`IIP`) function.
- Finally, our method gives an elementary and short proof for the mentioned exponential D^{EQ} lower bound of Chattopadhyay, Lovett, and Vinyals for Integer Inner Product (`IIP`).

2012 ACM Subject Classification Theory of computation → Communication complexity; Theory of computation → Oracles and decision trees

Keywords and phrases Boolean function complexity, parity decision trees, randomized communication complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.37

Funding *Hamed Hatami*: Supported by an NSERC grant.
Kaave Hosseini: Partially supported by the Goergen Institute for Data Science at the University of Rochester.
Morgan Shirley: Research was done while the author was a student at the University of Toronto. Supported by an NSERC grant.



© Tsun-Ming Cheung, Hamed Hatami, Kaave Hosseini, Aleksandar Nikolov, Toniann Pitassi, and Morgan Shirley;
licensed under Creative Commons License CC-BY 4.0



1 Introduction

We start by recalling two fundamental notions related to the complexities of Boolean functions: the *randomized parity decision tree complexity* and the *spectral norms*.

A *parity decision tree* (PDT) for a Boolean function $f : \mathbb{F}_2^n \to \{0,1\}$ is similar to a standard decision tree with the following strengthened query power. Instead of a single variable, each internal node can query the parity (sum over $\mathbb{F}_2$) of any subset of variables, e.g., $x_1 \oplus x_5 \oplus x_6$. The *parity decision tree complexity* of f, denoted by $\mathsf{PDT}_{\mathrm{depth}}(f)$, is the smallest depth of a PDT that outputs the correct value of $f(x)$ on every input x. A *randomized parity decision tree* (RPDT) of depth at most d is a probability distribution over PDTs of depth at most d. It computes f with error ϵ if, for every input x, the RPDT outputs $f(x)$ with probability at least $1-\epsilon$. The *randomized parity decision tree complexity*, denoted by $\mathsf{RPDT}_{\mathrm{depth}}(f)$, is the smallest depth of an RPDT computing f with error $\epsilon = 1/3$.

Parity decision trees are closely related to the Fourier expansions of Boolean functions. The *spectral norm* (also known as *algebra norm*) of a function $f : \mathbb{F}_2^n \to \mathbb{R}$, denoted by $\|\widehat{f}\|_1$, is the sum of the absolute values of the Fourier coefficients.

It is easy to see from the Fourier expansion of a PDT that every Boolean function $f : \mathbb{F}_2^n \to \{0,1\}$ satisfies $\log \|\widehat{f}\|_1 \leq \mathsf{PDT}_{\mathrm{depth}}(f)$. The randomized counterpart of this inequality does not hold as illustrated by h, the indicator function of the standard basis vectors $\{e_1, \ldots, e_n\}$: as observed in [14], $\mathsf{RPDT}_{\mathrm{depth}}(h) = O(1)$, but $\log \|\widehat{h}\|_1 = \Theta(\log n)$. Nonetheless, this still leaves the possibility of $\log \|\widehat{f}\|_1 = \widetilde{O}(\mathsf{RPDT}_{\mathrm{depth}}(f))$ open, where $\widetilde{O}(\cdot)$ hides polylogarithmic dependencies on n. Cheung et al. [8] specifically asked whether such a relation could hold, which if confirmed, would have established the stronger lower bound on $\mathsf{RPDT}_{\mathrm{depth}}$.

The main result of this work gives an example in which $\log \|\widehat{f}\|_1$ is exponential in $\mathsf{RPDT}_{\mathrm{depth}}(f)$, answering the question by [8] in the negative.

▶ **Theorem 1** (Main theorem). *There exists a function $f : \mathbb{F}_2^n \to \{0,1\}$ with*

$$\mathsf{RPDT}_{\mathrm{depth}}(f) = O(\log n) \quad \textit{and} \quad \log \|\widehat{f}\|_1 = \Omega(n).$$

In the converse direction, the quadratic gap $\mathsf{RPDT}_{\mathrm{depth}}(f) = O(\|\widehat{f}\|_1^2)$ holds for every Boolean function (see e.g. [14, Lemma 2.7]). Chattopadhyay, Mande, and Sherif [5] proved that the sink function SINK satisfies $\mathsf{RPDT}_{\mathrm{depth}}(\mathsf{SINK}) = \Omega(\|\widehat{\mathsf{SINK}}\|_1)$. This result and Theorem 1 together demonstrate that the measures $\log \|\widehat{f}\|_1$ and $\mathsf{RPDT}_{\mathrm{depth}}(f)$ are not polynomially related in both ways.

The proof technique used in the main theorem involves an application of Hölder's inequality with carefully-chosen parameters (see Section 3) that allow for a combinatorial interpretation of the problem. This framework is useful not only for the question considered above but also in communication complexity, where we will use the Hölder's inequality technique to simplify the proofs of existing lower bounds. For the rest of this section, we will discuss applications of both the proof technique and of the main theorem itself.

1.1 Communication Complexity: The Power of Oracle Access to Equality

Arguably the most well-known communication problem is the Equality function EQ, in which the two parties compare if their inputs are identical. In the model of *public-coin randomized communication*, two parties are given a publicly accessible random string and are

allowed to make errors with probability bounded away from 1/2. It is an elementary result that $\text{EQ} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ has a randomized protocol requiring only $O(1)$ bits of communication, while any deterministic protocol of EQ requires $n+1$ bits of communication.

A natural question would be whether Equality fully captures the power of randomness in communication. A formal formulation of the question is to compare the relative power of randomized protocols and *deterministic protocols with oracle access to* Equality or, in short, D^{EQ} protocols. A D^{EQ} protocol performs communications as usual; in addition, the parties are given access to an oracle that computes Equality function, and each oracle call is charged at a cost of one bit. The seminal work by Chattopadhyay, Lovett and Vinyals [4] considered this question and proved an exponential separation between the D^{EQ} complexity and randomized communication complexity (denoted as R) of the *Integer Inner Product* function ($\texttt{IIP}$) in dimension at least 6.

For a chosen dimension k and parameter n, the Boolean matrix $\texttt{IIP}_k^{(n)} : \{-2^n, \ldots, 2^n\}^k \times \{-2^n, \ldots, 2^n\}^k \to \{0,1\}$ is defined by

$$\texttt{IIP}_k[(x_1, \ldots, x_k), (y_1, \ldots, y_k)] = \begin{cases} 1 & \text{if } \sum_{i=1}^k x_i y_i = 0 \\ 0 & \text{otherwise} \end{cases}.$$

For this communication problem, each player holds a $\Theta(kn)$-bit input. As we only consider the case when $k = O(1)$, we treat $\texttt{IIP}_k^{(n)}$ as a $\Theta(n)$-bit communication problem.

Chattopadhyay, Lovett, and Vinyals presented a (one-way) randomized protocol for $\texttt{IIP}_k^{(n)}$ of cost $O(k \log n)$. For the lower bound, they introduced a lower bound technique which we call *relative area method* (see Theorem 13), and showed that no D^{EQ} protocol could compute the function with fewer than $\Omega(n)$ bits of communication.

▶ **Theorem 2** ([4]). *For constant $k \geq 6$, $\mathsf{R}\left(\texttt{IIP}_k^{(n)}\right) = O(\log n)$ and $\mathsf{D}^{\mathsf{EQ}}\left(\texttt{IIP}_k^{(n)}\right) = \Omega(n)$.*

In a subsequent work, Cheung et al. [8] obtained a strengthening of Theorem 2 via a different approach of spectral methods. It was shown in [14] that

$$\mathsf{D}^{\mathsf{EQ}}(A) \geq \frac{1}{2} \log \|A\|_{\text{ntr}} \tag{1}$$

for any boolean matrix $A \in \{0,1\}^{m \times n}$, where $\|A\|_{\text{ntr}}$ is the *normalized trace norm* of A with the normalization factor from the matrix dimensions (see Definition 10 for the precise definition).

Cheung et al. [8] observed that for $k \geq 3$, the $\Theta(2^{kn}) \times \Theta(2^{kn})$ matrix $\texttt{IIP}_k^{(n)}$ contains a $2^{4n/3} \times 2^{4n/3}$ *point-line incidence* matrix $\texttt{PL}^{(4n/3)}$ as a submatrix. The matrix $\texttt{PL}^{(n)}$ is defined on the domains $\mathcal{P} = \mathcal{L} = [2^{n/4}] \times [2^{3n/4}] \subseteq \mathbb{Z}^2$, and the entries of $\texttt{PL}^{(n)} : \mathcal{P} \times \mathcal{L} \to \{0,1\}$ are given by

$$\texttt{PL}[(x,x'),(y,y')] = \begin{cases} 1 & \text{if } xy + x' = y' \\ 0 & \text{otherwise} \end{cases}.$$

They proved that the normalized trace norm of $\texttt{PL}^{(n)}$ is exponential in n, which shows that $\mathsf{D}^{\mathsf{EQ}}\left(\texttt{PL}^{(n)}\right) = \Omega(n)$. Since communication complexity does not increase under restriction, this subsequently implies Theorem 2.

▶ **Theorem 3** ([8]). *The $2^m \times 2^m$ Boolean matrix $\texttt{PL}^{(m)}$ satisfies $\|\texttt{PL}^{(m)}\|_{\text{ntr}} = \Omega(2^{m/32})$. Consequently, for $k \geq 3$,*

$$\mathsf{D}^{\mathsf{EQ}}\left(\texttt{IIP}_k^{(n)}\right) \geq \frac{n}{48} + O(1).$$

A common shortcoming of both proofs of Theorem 2 in [4, 8] is their highly technical nature. In Section 4, we give a considerably simplified proof – with improved linear factor on $\mathsf{D}^{\mathsf{EQ}}\left(\mathtt{IIP}_k^{(n)}\right)$ – based on the Hölder's inequality technique. Here, we consider a different submatrix of $\mathtt{IIP}_k$, which is the point-line incidence system $\mathtt{PL}_*$ that proves the optimality of Szemerédi–Trotter theorem. We will provide the precise definition of $\mathtt{PL}_*^{(m)}$ in Section 4.

▶ **Theorem 4** (Improved version of Theorem 3). *The $2^{m+1} \times 2^m$ Boolean matrix $\mathtt{PL}_*^{(m)}$ satisfies $\|\mathtt{PL}_*^{(m)}\|_{\mathrm{ntr}} = \Omega(2^{m/6})$. Consequently, for $k \geq 3$,*

$$\mathsf{D}^{\mathsf{EQ}}\left(\mathtt{IIP}_k^{(n)}\right) \geq \frac{n}{8} + O(1).$$

Nondeterministic communication

When analyzing the power of Equality in communication, another model of interest is *nondeterministic protocols with oracle access to* Equality, or N^{EQ} for short. For the precise definition of the N^{EQ} model, we refer readers to [19]. N^{EQ} and its relationships with related models have been studied previously, both implicitly in [12] and explicitly in [19]. In the latter work, Pitassi, Shirley and Shraibman analyzed the relative area method used by [4] to lower bound $\mathsf{D}^{\mathsf{EQ}}(\mathtt{IIP}_6)$ and showed that the same technique is applicable to N^{EQ} complexity as well. It is not clear that the lower bound technique of [8] on the D^{EQ} complexity of $\mathtt{IIP}_3$ works for N^{EQ}, so prior to this work it was open whether $\mathtt{IIP}_3$ is hard for N^{EQ}. We observe that the analysis in Theorem 4 can be adapted to the relative area method, and therefore yields an almost linear lower bound on $\mathsf{N}^{\mathsf{EQ}}(\mathtt{IIP}_k)$ for every $k \geq 3$.

▶ **Theorem 5.** *For constant $k \geq 3$,*

$$\mathsf{N}^{\mathsf{EQ}}\left(\mathtt{IIP}_k^{(n)}\right) = \Omega\left(\frac{n}{\log n}\right).$$

As far as we know, this method cannot be improved to give a linear lower bound on $\mathsf{N}^{\mathsf{EQ}}(\mathtt{IIP}_3)$. We leave resolving this logarithmic gap as an open question.

XOR-lifts

A related open problem posed in [8] is whether an exponential separation of D^{EQ} and $\mathbb{R}$ holds for an XOR-lift. For a Boolean function $f : \mathbb{F}_2^n \to \{0,1\}$, the XOR-lift $f^{\oplus} : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \{0,1\}$ is defined by $f^{\oplus}(x,y) = f(x \oplus y)$ for each $x, y \in \mathbb{F}_2^n$. It can be verified that $\mathtt{IIP}_k$ is not an XOR-lift, so the D^{EQ}-vs-R separation for the XOR-lift case was open.

The matrix class of XOR-lifts is interesting in complexity theory since many complexity measures of the matrix $f^{\oplus}$ can be related (or even equated) to complexity measures of the Boolean function f. Indeed, the query-to-communication connections allow us to translate the separation question to a purely query-complexity setting. For any Boolean function $f : \mathbb{F}_2^n \to \{0,1\}$, the inequality

$$\mathsf{R}(f^{\oplus}) \leq 2\,\mathsf{RPDT}_{\mathrm{depth}}(f) \tag{2}$$

is evident from the standard simulation of a randomized parity decision tree by a communication protocol. A result of [10] shows that

$$\|f^{\oplus}\|_{\mathrm{ntr}} = \|\widehat{f}\|_1. \tag{3}$$

Immediate from Equations (1)–(3), we obtain the exponential D^{EQ}-vs-R separation for XOR-lifts from Theorem 1.

▶ **Corollary 6.** *There exists a Boolean function* $f : \mathbb{F}_2^n \to \{0,1\}$ *such that its* XOR*-lift* $f^{\oplus}$ *satisfies* $\mathsf{R}(f^{\oplus}) = O(\log n)$ *and* $\mathsf{D}^{\mathsf{EQ}}(f^{\oplus}) = \Omega(n)$.

1.2 Query Complexity: The Power of Randomness

Understanding the power of randomized versus deterministic algorithms is a fundamental problem in complexity theory. Depending on the computational model, this problem varies from fully resolved to forbiddingly out of reach. In the standard decision tree model, it is well known that randomness does not significantly reduce the number of queries. One early result in complexity theory by Nisan [18] showed that a randomized decision tree of depth d computing a Boolean function can be simulated by a deterministic decision tree of depth at most $O(d^3)$. On the other end of the spectrum, the randomized-versus-deterministic problem remains overwhelmingly difficult for Turing Machines.

A generic framework to study the relative powers of two computation models is to study the relations of the complexity classes defined by suitable complexity measures. For the sake of comparisons between measures, we consider the complexity measures normalized to the range $[0, n]$. Given a complexity measure $C(\cdot)$ for a deterministic computation model, $\mathtt{P}$ is defined to be the class of functions $f : \mathbb{F}_2^n \to \{0,1\}$ (more accurately, sequences of functions f_n) such that $C(f) \leq \text{polylog}(n)$. We define the class $\mathtt{BPP}$ similarly for the randomized counterpart of $C(\cdot)$. The inclusion $\mathtt{P} \subseteq \mathtt{BPP}$ for every measure is obvious, so the randomized-versus-deterministic problem amounts to whether the classes $\mathtt{P}$ and $\mathtt{BPP}$ are equal.

We consider the two natural measures for trees, namely depth and logarithmic size (for normalization purposes) and four types of decision trees: (standard) decision tree (DT), parity decision tree (PDT), randomized decision tree (RDT), and randomized parity decision tree (RPDT). This branches into eight complexity measures in concern, and we define the other six measures $\mathsf{DT}_{\text{depth}}$, $\log \mathsf{DT}_{\text{size}}$, $\mathsf{RDT}_{\text{depth}}$, $\log \mathsf{RDT}_{\text{size}}$, $\mathsf{PDT}_{\text{depth}}$ and $\log \mathsf{PDT}_{\text{size}}$ in an analogous manner to $\mathsf{RPDT}_{\text{depth}}$ and $\log \mathsf{RPDT}_{\text{size}}$. Based on the type of queries equipped and the complexity measure, the P-vs-BPP question branches into four pairs for comparison.

Standard decision trees

Nisan [18] showed that $\mathsf{DT}_{\text{depth}}(f) \leq O(\mathsf{RDT}_{\text{depth}}(f)^3)$ in 1988 and settled that $\mathtt{P} = \mathtt{BPP}$ in the depth setting. The question in the size setting remained unsettled for decades until recently Chattopadhyay, Dahiya, Mande, Radhakrishnan, and Sanyal [3] showed that

$$\log \mathsf{DT}_{\text{size}}(f) = O\left(\log^4(\mathsf{RDT}_{\text{size}}(f)) \log^3 n\right),$$

concluding that $\mathtt{P} = \mathtt{BPP}$ in this setting as well.

Parity decision trees

In the depth setting, the two classes $\mathtt{P}$ and $\mathtt{BPP}$ are strongly separated by the simple function of OR. It is well-known that the OR function on n bits satisfies $\mathsf{RPDT}_{\text{depth}}(\mathsf{OR}) = O(1)$ but $\mathsf{PDT}_{\text{depth}}(\mathsf{OR}) = n$, which provides the optimal separation and hence $\mathtt{P} \neq \mathtt{BPP}$ in this setting.

Note that however $\mathsf{PDT}_{\text{size}}(\mathsf{OR}) = O(n)$, so OR does not attest a separation between $\mathtt{P}$ and $\mathtt{BPP}$ in the size setting. Indeed, previous to this work the P-vs-BPP question was unknown in the size setting. As mentioned that $\log \|\widehat{f}\|_1 \leq \mathsf{PDT}_{\text{depth}}(f)$ for every function f, an immediate corollary of Theorem 1 implies that $\mathtt{P} \neq \mathtt{BPP}$ in the size setting for parity decision trees.

▶ **Corollary 7.** *There is a function* $f : \mathbb{F}_2^n \to \{0,1\}$ *such that* $\log \mathsf{RPDT}_{\text{size}}(f) = O(\log n)$, *but* $\log \mathsf{PDT}_{\text{size}}(f) = \Omega(n)$.

In fact, the randomized parity decision tree in our construction is non-adaptive and has a one-sided error, hence we obtain the stronger separation $\mathtt{P} \neq \mathtt{RP}$[1] in the size setting parity decision tree size model. It remains an interesting open problem to determine whether a separation of $O(1)$-vs-$\Omega(n)$ for $\log \mathsf{RPDT}_{\text{size}}$ and $\log \mathsf{PDT}_{\text{size}}$ is possible, as in the case of depth illustrated by the OR function.

▶ **Question 8.** *Is there a boolean function* $f : \mathbb{F}_2^n \to \{0,1\}$ *such that* $\log \mathsf{RPDT}_{\text{size}}(f) = O(1)$, *but* $\log \mathsf{PDT}_{\text{size}}(f) = \Omega(n)$, *or even* $\log \mathsf{PDT}_{\text{size}}(f) = \omega(\log n)$*?*

1.3 Fourier Analysis: Approximate versus Exact Spectral Norms

The notion of *approximate norms* is customary in complexity theory as complexity measures for models with error tolerance. The *approximate spectral norm* of a function f with error ϵ, denoted as $\widehat{L}_{1,\epsilon}(f)$[2], is the minimum $\|\widehat{g}\|_1$ for some function g such that $\|f - g\|_\infty \leq \epsilon$. As in the case of other complexity measures, we adopt the canonical choice $\epsilon = 1/3$ when referring to constant error. Spectral norm and approximate spectral norm of a Boolean function are fundamental parameters that find applications in many areas such as learning theory [17], complexity theory [21, 22, 23, 1], communication complexity [23, 14, 8], Fourier analysis [23, 11] and additive combinatorics [13, 20, 9].

It is natural to ask whether the spectral norm of a Boolean function is upper bounded by its approximate spectral norm. Towards answering this question, Cheung et al. [9] observed that if f is the indicator function of n-bit strings of Hamming-weight 1, then $\widehat{L}_{1,1/3}(f) = O(1)$ but $\|\widehat{f}\|_1 = \Omega(\sqrt{n})$. Nevertheless, this separation does not rule out the possibility of polynomial relations with dependency on n such as $\|\widehat{f}\|_1 = \text{poly}\left(\widehat{L}_{1,1/3}(f), n\right)$. Using the result by [14] that $\log \|\widehat{f}\|_1 \leq \mathsf{RPDT}_{\text{depth}}(f)$ for any Boolean function f, Theorem 1 immediately rules out the possibility of polynomial dependency.

▶ **Corollary 9.** *There exists a function* $f : \mathbb{F}_2^n \to \{0,1\}$ *such that* $\widehat{L}_{1,1/3}(f) = n^{O(1)}$ *but* $\|\widehat{f}\|_1 = 2^{\Omega(n)}$.

2 Preliminaries

For a positive integer k, we denote $[k] \coloneqq \{1, \ldots, k\}$. All logarithms in this paper are base 2.

We denote the indicator function of a predicate P as $\mathbf{1}[P]$ and the indicator function of a set S as $\mathbf{1}_S$. For a random variable r and a set S, we write $r \sim S$ to indicate that r is uniformly sampled from S. Throughout this work, we adopt the standard big-O notations in computer science.

2.1 Schatten norms

For a vector $v \in \mathbb{R}^k$ and $p \in [1, \infty]$, we denote the ℓ_p-norm of v as $\|v\|_p$. For a matrix $A \in \mathbb{R}^{m \times n}$, the singular values of A are the square roots of the eigenvalues of $AA^\top$, which we denote by $\sigma_1(A) \geq \sigma_2(A) \geq \ldots \geq \sigma_{\min\{m,n\}}(A) \geq 0$. The central matrix norms in this paper are *Schatten p-norm*, which is defined as

[1] RP denotes the class of functions computable by RPDT with constant one-sided error and poly-logarithmic costs

[2] We adopt this function-like notation to emphasize that approximate spectral norm is not a norm despite what the name suggests

$$\|A\|_{S_p} = \left(\sum_i \sigma_i(A)^p\right)^{1/p}$$

for $p \in [1,\infty)$, and $\|A\|_{S_\infty} = \sigma_1(A)$. Put differently, Schatten p-norm of a matrix is the ℓ_p norm of the vector $\big[\sigma_1(A), \sigma_2(A), \ldots, \sigma_{\min\{m,n\}}(A)\big]$. One property of Schatten norms inherited from ℓ_p norms is the monotonicity of Schatten p-norm in p: $\|A\|_{S_p} \geq \|A\|_{S_q}$ for $1 \leq p < q \leq \infty$.

The Schatten 1-norm, 2-norm and ∞-norm are frequently used and are commonly known as *trace norm*, *Frobenius norm*, and *spectral norm* respectively:

$$\|A\|_{\mathrm{tr}} = \|A\|_{S_1} = \sum_i \sigma_i(A);$$

$$\|A\|_{\mathrm{F}} = \|A\|_{S_2} = \sqrt{\sum_i \sigma_i(A)^2} = \sqrt{\sum_{ij} A_{ij}^2};$$

$$\|A\| = \|A\|_{S_\infty} = \sigma_1(A) = \max_{x \in \mathbb{R}^n \setminus \{0\}} \frac{\|Ax\|_2}{\|x\|}.$$

For applications in theoretical computer science, it is more convenient to work with the *normalized trace norm* as a complexity measure.

▶ **Definition 10** (Normalized trace norm). *For $A \in \mathbb{R}^{m\times n}$, the normalized trace norm of A is*

$$\|A\|_{\mathrm{ntr}} \coloneqq \frac{\|A\|_{\mathrm{tr}}}{\sqrt{mn}} = \frac{1}{\sqrt{mn}} \sum_i \sigma_i(A).$$

We remind readers that unlike essentially all other complexity measures and matrix norms used in this work, the normalized trace norm may increase upon restriction to a submatrix.

2.2 Fourier analysis of Boolean functions

This section gives a basic overview of Fourier analysis on the Boolean cube. For every $\eta \in \mathbb{F}_2^n$, define the Fourier character $\chi_\eta : \mathbb{F}_2^n \to \{\pm 1\}$ as $\chi_\eta(x) = (-1)^{\langle x,\eta\rangle}$, where $\langle x, \eta\rangle$ is the standard inner product over $\mathbb{F}_2^n$. For a function $f : \mathbb{F}_2^n \to \mathbb{R}$, its Fourier expansion is given by

$$f = \sum_{\eta \in \mathbb{F}_2^n} \widehat{f}(\eta)\chi_\eta, \tag{4}$$

where the Fourier coefficient $\widehat{f}(\eta)$ is defined as $\widehat{f}(\eta) \coloneqq \mathbb{E}_{x\sim\mathbb{F}_2^n}[f(x)\chi_\eta(x)]$.

For a function $f : \mathbb{F}_2^n \to \mathbb{R}$ and $p \in [1,\infty)$, the Fourier p-norm is defined as

$$\|\widehat{f}\|_p = \left(\sum_{x\in\mathbb{F}_2^n} |\widehat{f}(\eta)|^p\right)^{1/p},$$

and $\|\widehat{f}\|_\infty = \max_\eta |\widehat{f}(\eta)|$. In other words, the Fourier p-norm is the ℓ_p norm of the vector of Fourier coefficients. Fourier 1-norm is better known as the *spectral norm* of f, i.e. $\|\widehat{f}\|_1 \coloneqq \sum_{\eta\in\mathbb{F}_2^n} |\widehat{f}(\eta)|$.

For an Abelian group, the convolution of two *real-valued* functions $f, g : G \to \mathbb{R}$ is a function defined to be

$$f * g(x) \coloneqq \mathop{\mathbb{E}}_{y\sim G}[f(y)g(x-y)].$$

It is a standard fact in Fourier analysis that the Fourier transformation of convolution of two functions is the pointwise product of their respective Fourier transforms: $\widehat{f * g} = \widehat{f} \cdot \widehat{g}$.

Lastly, *Parseval's identity* states that the squared Fourier 2-norm of a function f is equal to its second moment (under uniform probability measure):

$$\|\widehat{f}\|_2^2 = \sum_{\eta \in \mathbb{F}_2^n} \widehat{f}(\eta)^2 = \mathop{\mathbb{E}}_{x \sim \mathbb{F}_2^n}[f(x)^2].$$

3 The Proof Framework

The key proof technique throughout this work is Hölder's inequality with a nuanced choice of p-norms. Hölder's inequality states that for any $p, q \in [1, \infty]$ satisfying $\frac{1}{p} + \frac{1}{q} = 1$,

$$|\langle u, v \rangle| \leq \|u\|_p \|v\|_q \tag{5}$$

for any real vectors $u, v \in \mathbb{R}^n$. A generalization of Hölder's inequality known as *Littlewood's inequality* [15], which can be deduced from the standard Hölder's inequality, relates the p-norms of a function for some "interpolated" tuples of p. The inequality states that for $1 \leq p_0 < p_* < p_1 \leq \infty$,

$$\|v\|_{p_*} \leq \|v\|_{p_0}^{\theta} \|v\|_{p_1}^{1-\theta} \tag{6}$$

for any vector $v \in \mathbb{R}^n$, where $\theta \in (0, 1)$ satisfies $\frac{1}{p_*} = \frac{\theta}{p_0} + \frac{1-\theta}{p_1}$. As Schatten norms and Fourier norms are defined based on ℓ_p norms, the above inequality also holds with replacing the ℓ_p norms with these families of norms.

Setting the parameters $(p_*, p_0, p_1) = (2, 1, 4)$ (so that $\theta = 1/3$) for Equation (6) yields the following lower bound for 1-norm (after some algebraic manipulations):

$$\|f\|_1 \geq \frac{\|f\|_2^3}{\|f\|_4^2}. \tag{7}$$

As this is the sole parametrization of Hölder's inequality that we employ, we colloquially refer to Equation (7) as "the Hölder's inequality" in the rest of this work.

Among all the possible parametrizations for Littlewood's inequality, the choice made in Equation (7) of $(p_*, p_0, p_1) = (2, 1, 4)$ is appealing because of the nice physical interpretations of 2-norm and 4-norm for Boolean matrices (the Schatten norms) and Boolean functions (the Fourier norms). One main inspiration of Equation (7) as a practical bound is the work of Chazelle and Lvov on hereditary discrepancy [7]. Chazelle and Lvov proved a hereditary discrepancy lower bound of a matrix in terms of its Schatten 2-norm and 4-norm, highlighting the combinatorial interpretations of these norms for Boolean matrices. In the next section, we illustrate the combinatorial perspective of Equation (7) with a simpler proof of Theorem 3.

4 Simplified Proof of Communication Complexity Lower Bounds

It is customary to associate a Boolean matrix A with the biadjacency matrix of a bipartite graph $G = (U \cup V, E)$. In view of this, the square of Schatten 2-norm i.e. Frobenius norm of A is simply the number of edges of G. The Schatten 4-norm of A also admits a graph-theoretic interpretation:

$$\|A\|_{S_4}^4 = \mathrm{Tr}((A^\top A)^2) = \sum_{i,j \in U} \sum_{k,\ell \in V} A_{ik} A_{i\ell} A_{jk} A_{j\ell} = \sum_{i,j \in U} \sum_{k,\ell \in V} \mathbf{1}[\{ik, i\ell, jk, j\ell\} \subseteq E].$$

In other words, $\|A\|_{S_4}^4$ is precisely the count of possibly degenerate 4-cycles in G. This quantity becomes especially easy to evaluate when the underlying graph G does not contain non-degenerate 4-cycles (C_4-free), or equivalently, A does not contain a 2×2 all-one submatrix.

For a C_4-free bipartite graph, all contributions to $\|A\|_{S_4}^4$ come from the counts of edges and paths of length 2. Furthermore, if G is almost balanced in the sense that the average degree of G is close to the maximum degree, Equation (7) indeed yields a non-trivial lower bound for $\|A\|_{\mathrm{ntr}}$:

▶ **Proposition 11.** *Let $G = (U \cup V, E)$ be a C_4-free bipartite graph with maximum degree $\Delta(G)$. For $x \in U \cup V$, let d_x be the degree of x. If $A \in \{0,1\}^{U\times V}$ is the biadjacency matrix of G, Then*

$$\|A\|_{\mathrm{tr}} \geq \frac{|E|^{3/2}}{\sqrt{\sum_{x\in U\cup V} d_x^2}} \geq \frac{|E|}{\sqrt{2\Delta(G)}}.$$

Proof. By Hölder's inequality (Equation (7)),

$$\|A\|_{\mathrm{tr}} \geq \frac{\|A\|_{\mathrm{F}}^3}{\|A\|_{S_4}^2} = \frac{|E|^{3/2}}{\sqrt{\mathrm{Tr}((A^\top A)^2)}}.$$

Since G is C_4-free, the expression of $\mathrm{Tr}((A^\top A)^2)$ is simplified to

$$\begin{aligned}\mathrm{Tr}((A^\top A)^2) &= \sum_{i\in U}\sum_{k\in V} A_{ik} + \sum_{i\in U}\sum_{\substack{k,\ell\in V\\ k\neq\ell}} A_{ik}A_{i\ell} + \sum_{\substack{i,j\in U\\ i\neq j}}\sum_{k\in V} A_{ik}A_{jk}\\ &= |E| + \sum_{i\in U} d_i(d_i-1) + \sum_{k\in V} d_k(d_k-1)\\ &= \left(\sum_{x\in U\cup V} d_x^2\right) - |E|\end{aligned}$$

and the required bounds follow. ◀

Since every two distinct points define a unique line, the bipartite graph G associated with any point-line incidence system (with no duplicated lines) is C_4-free. From Proposition 11, one can readily derive an improved exponential normalized trace norm lower bound for $\mathtt{PL}$ using the same point-incidence submatrix in [8].

We prove a better bound by considering a point-line incidence system $\mathtt{PL}_*$ attributed to Paul Erdős (see [6, Lemma 6.25]). This point-line incidence system is constructed to show the tightness of the Szemerédi–Trotter incidence bound. Concretely, the $2^{n+1}\times 2^n$ matrix $\mathtt{PL}_*^{(n)} : \mathcal{P}'\times\mathcal{L}' \to \{0,1\}$ is defined as follows: the input domains are $\mathcal{P}' = [2^{n/3}]\times[2^{2n/3+1}]$ and $\mathcal{L}' = [2^{n/3}]\times[2^{2n/3}]$, and the entries are given by

$$\mathtt{PL}_*^{(n)}[(x,x'),(y,y')] = \begin{cases} 1 & \text{if } xy + y' = x'\\ 0 & \text{otherwise}\end{cases}.$$

We first show that $\mathtt{IIP}_3$ contains $\mathtt{PL}_*$ as a submatrix.

▷ Claim 12. The matrix $\mathtt{IIP}_3^{(n)}$ contains the matrix $\mathtt{PL}_*^{(3n/2-3/2)}$ as a submatrix.

Proof. The condition of incidence of $\mathtt{PL}_*$ can be rewritten as $xy + (1)y' + x'(-1) = 0$, which is an instance of integer inner product. For $\mathtt{PL}_*^{(3n/2-3/2)}$, the magnitude of each entry is bounded by $2^{\frac{2}{3}\left(\frac{3}{2}n-\frac{3}{2}\right)+1} = 2^n$, therefore $\mathtt{PL}_*^{(3n/2-3/2)}$ is a submatrix of $\mathtt{IIP}_3^{(n)}$. ◁

Proof of Theorem 4. As noted above, $\mathtt{PL}_*^{(m)}$ is C_4-free because no two distinct lines intersect at the same pair of points. For the matrix $\mathtt{PL}_*^{(m)}$, each line is incident to exactly one point of the form $(i, x') \in \mathcal{P}$ for each $i \in [2^{m/3}]$. Therefore the bipartite graph defined by $\mathtt{PL}_*$ contains $|\mathcal{L}| \cdot 2^{m/3} = 2^{4m/3}$ edges. Also, each point is incident to at most one line of the form $(j, y') \in \mathcal{L}$ for each $j \in [2^{m/3}]$, hence the maximum degree of the graph is $2^{m/3}$. By Proposition 11,

$$\left\|\mathtt{PL}_*^{(m)}\right\|_{\mathrm{ntr}} \geq \Omega\left(\frac{1}{2^m} \cdot \frac{2^{4m/3}}{\sqrt{2 \cdot 2^{m/3}}}\right) = \Omega\left(2^{m/6}\right).$$

By Claim 12 and Equation (1), we obtain

$$\mathsf{D}^{\mathsf{EQ}}\left(\mathtt{IIP}_k^{(n)}\right) \geq \mathsf{D}^{\mathsf{EQ}}\left(\mathtt{IIP}_3^{(n)}\right) \geq \mathsf{D}^{\mathsf{EQ}}\left(\mathtt{PL}_*^{(3n/2-3/2)}\right) \geq \frac{1}{2}\left(\frac{1}{6} \cdot \frac{3n}{2}\right) + O(1) = \frac{n}{8} + O(1). \quad \blacktriangleleft$$

As mentioned earlier, Chattopadhyay, Lovett, and Vinyals [4] proved a linear lower bound on $\mathsf{D}^{\mathsf{EQ}}(\mathtt{IIP}_k)$ for $k \geq 6$ by a method different from the approach in [8] and this work. The method used in [4], the relative area method, considered a parametrized weighted sum of monochromatic rectangle partition of a Boolean matrix. The following D^{EQ} lower bound is deduced from the communication complexity lower bound with more general oracle access.

▶ **Theorem 13** ([4, Lemmas 3.5 and 3.7])**.** *Suppose M is a Boolean matrix with α 1-entries, and the maximum area of any 1-monochromatic rectangle in M is β. For any constant $\eta \in (1/2, 1)$,*

$$\mathsf{D}^{\mathsf{EQ}}(M) = \Omega\left(\log\left(\frac{\alpha}{\beta^{1-\eta}|M|^{\eta}}\right)\right).$$

For the matrix $\mathtt{PL}_*^{(m)}$, as observed in the proof of Theorem 4, $\alpha = |E(G)| = 2^{4m/3}$ and $\beta = \Delta(G) = 2^{m/3}$ since $\mathtt{PL}_*^{(m)}$ is C_4-free. Applying Theorem 13 with $\eta = 1/2 + \epsilon'$ for some very small constant $\epsilon' > 0$, we obtain a linear D^{EQ} lower bound with a slightly inferior constant due to the slackness in η. We can also use this technique to lower bound N^{EQ}, which proves Theorem 5.

▶ **Theorem 14** ([19])**.** *Let M be a $2^m \times 2^m$ Boolean matrix, α be the number of 1-entries in M and let β be the size of the largest 1-monochromatic rectangle in M. Then*

$$\mathsf{N}^{\mathsf{EQ}}(M) = \Omega\left(\log\left(\frac{\alpha}{\sqrt{\beta} 2^m}\right) \cdot \frac{1}{\log m}\right).$$

Proof of Theorem 5. As noted before, $\alpha = |E(G)| = 2^{4m/3}$ and $\beta = \Delta(G) = 2^{m/3}$. For $m = 3n/2 - 3/2$, applying Theorem 14 gives

$$\mathsf{N}^{\mathsf{EQ}}\left(\mathtt{PL}_*^{(m)}\right) = \Omega\left(\frac{n}{\log n}\right)$$

and therefore the same lower bound holds for $\mathsf{N}^{\mathsf{EQ}}\left(\mathtt{IIP}_k^{(n)}\right)$ for $k \geq 3$. ◀

5 Proof of the Main Theorem

The Hölder's inequality supplies a large lower bound on 1-norm in the scenario of large 2-norm and small 4-norm. The proof of Theorem 3 utilizes the C_4-free property and sufficient edge density of $\mathtt{PL}$ to show the desired trace norm lower bound. We adopt the same strategy in exhibiting a function that the separation of randomized parity decision tree depth and spectral norm.

As in the matrix case, we begin by examining the physical interpretations of p-norms in the Hölder's inequality. A fundamental quantity in additive combinatorics known as the *additive energy*, coincides with the fourth power of Fourier 4-norm in Boolean function setting.

▶ **Definition 15** (Additive energy). *For $f : G \to \{0,1\}$ over an Abelian group G, the additive energy of f is defined as*

$$E(f) \coloneqq \mathop{\mathbb{E}}_{x,y,z\sim G}[f(x)f(y)f(z)f(x+y-z)].$$

For a Boolean function $f : \mathbb{F}_2^n \to \{0,1\}$, tt is straightforward to show that $E(f) = \|\widehat{f}\|_4^4$:

$$\begin{aligned} E(f) &= \mathop{\mathbb{E}}_{t\sim\mathbb{F}_2^n}\left[\mathop{\mathbb{E}}_{x\sim\mathbb{F}_2^n}[f(x)f(t-x)] \cdot \mathop{\mathbb{E}}_{z\sim\mathbb{F}_2^n}[f(z)f(t-z)]\right] \\ &= \mathop{\mathbb{E}}_{t\sim\mathbb{F}_2^n}[(f*f(t))^2] \\ &= \sum_{\eta\in\mathbb{F}_2^n} \widehat{f*f}(\eta)^2 = \sum_{\eta\in G} \widehat{f}(\eta)^4 = \|\widehat{f}\|_4^4. \end{aligned}$$

It is also direct from Parseval's identity that $\|\widehat{f}\|_2^2 = \mathbb{E}_x[f(x)]$. Therefore in the Boolean function setting, Equation (7) states that

$$\|\widehat{f}\|_1 \geq \frac{\|\widehat{f}\|_2^3}{\|\widehat{f}\|_4^2} = \sqrt{\frac{\mathbb{E}_x[f(x)]^3}{E(f)}}. \tag{8}$$

Emulating the approach in Section 4 to guarantee a small 4-norm, we consider a Boolean function $f : \mathbb{F}_2^n \to \{0,1\}$ that satisfies $f(x)f(y)f(z)f(x+y+z) = 0$ for any distinct $x, y, z \in \mathbb{F}_2^n$ as an analogue of C_4-free bipartite graphs. For such a function, the additive energy of f is lower than typical as only $O(|\mathbb{F}_2^n|^2)$ tuples of $(x,y,z) \in (\mathbb{F}_2^n)^3$ could contribute to $E(f)$. A function with this property is precisely the indicator function of a Sidon set.

▶ **Definition 16** (Sidon set). *For an Abelian group G, a set $S \subseteq G$ is called a* Sidon *set if for every $a, b, c, d \in S$ such that $a + b = c + d$, then $\{a, b\} = \{c, d\}$.*

The indicator of a Sidon set emerges as a natural candidate function that attains the exponential spectral norm lower bound stated in Theorem 1. Indeed, one can show that the spectral norm of an indicator of a sufficiently big Sidon set is large.

▶ **Proposition 17.** *Every Sidon set S of a finite Abelian group G satisfies $\|\widehat{\mathbf{1}_S}\|_1 \geq \sqrt{|S|/2}$.*

Proof. Since S is a Sidon set, $\mathbf{1}_S(x)\mathbf{1}_S(y)\mathbf{1}_S(z)\mathbf{1}_S(x+y-z) = 1$ implies that $z = x$ or $z = y$. Thus

$$E(\mathbf{1}_S) \leq \frac{2}{|G|} \mathop{\mathbb{E}}_{x,y\sim G}[\mathbf{1}_S(x)\mathbf{1}_S(y)] = \frac{2}{|G|} \times \left(\frac{|S|}{|G|}\right)^2 = \frac{2|S|^2}{|G|^3}.$$

Applying Hölder's inequality (Equation (8)) gives

$$\|\widehat{\mathbf{1}_S}\|_1 \geq \sqrt{\frac{\mathbb{E}_x[\mathbf{1}_S(x)]^3}{E(\mathbf{1}_S)}} = \sqrt{\frac{|S|^3/|G|^3}{2|S|^2/|G|^3}} = \sqrt{\frac{|S|}{2}}. \qquad \blacktriangleleft$$

By Cauchy-Schwarz inequality and Parseval identity, it is direct to check that for any set $T \subseteq G$, the spectral norm of $\mathbf{1}_T$ is at most $\sqrt{|T|}$. The above proposition shows that this upper bound is tight up to constant factor for a Sidon set.

By considering the possible sums of a pair of elements, it is easy to see that for a Sidon set S, we have $\binom{|S|}{2} \leq |G|$ and hence $|S| = O(\sqrt{|G|})$. In the case of $G = \mathbb{Z}_2^n$ [3], a well-known construction of a Sidon set matching the size upper bound is the Bose–Chaudhuri–Hocquenghem (BCH) code [2, 16]. This code is constructed from polynomials over a finite field of characteristic 2.

Denote $\mathbb{F}_{2^m}$ the characteristic-2 finite field of size 2^m. The elements of $\mathbb{F}_{2^m}$ can be isomorphically identified with univariate $\mathbb{F}_2$-polynomials of degree at most $m-1$, where multiplication is defined modulus a fixed irreducible polynomial $P(x) \in \mathbb{F}_2[x]$ of degree m. Viewing as additive groups, $\mathbb{F}_{2^m} \cong \mathbb{F}_2[x]/\langle P(x)\rangle$ is isomorphic to $\mathbb{Z}_2^m$.

For $\mathbb{F}_p$ where p is a prime, the Sidon set construction in [2] is given by the tuples (a, a^k) over all a in a suitable subfield and a suitable exponent k. For $p = 2$, the construction takes the form of $\mathtt{BCH}(S)$, where for a set S we define

$$\mathtt{BCH}(S) \coloneqq \{(a, a^3) : a \in S\}.$$

For the sake of completeness, we include the proof for this specific construction.

▶ **Theorem 18** ([2]). *For every even number m, the set $\mathtt{BCH}(\mathbb{F}_{2^{m/2}}) \subseteq \mathbb{F}_{2^{m/2}} \times \mathbb{F}_{2^{m/2}} \equiv \mathbb{Z}_2^m$ is a Sidon set.*

Proof. Let (a, a^3) and (b, b^3) be two pairs with a prescribed sum $(u, v) \in \mathbb{F}_{2^{m/2}} \times \mathbb{F}_{2^{m/2}}$. This means $a + b = u$ and $a^3 + b^3 = v$, which implies that

$$u^3 = (a+b)^3 = a^3 + b^3 + (a+b)ab = v + uab.$$

Suppose $a \neq b$ i.e. $u \neq 0$, then $a(u+a) = ab = u^2 + vu^{-1}$. The same calculation concludes that a and b are the roots of the quadratic equation $x(x+u) = u^2 + vu^{-1}$. Since a quadratic equation has at most two roots, $\{a, b\}$ is uniquely determined by u and v. ◀

Theorem 18 allows us to construct a large Sidon set in $\mathbb{Z}_2^n$. Since a subset of a Sidon set remains a Sidon set, it remains to select a suitably structured subset whose indicator function is computable with a short randomized parity decision tree. As mentioned earlier, every element in $\mathbb{F}_{2^n}$ can be uniquely identified with a polynomial in $\mathbb{F}_2[x]$ of degree at most $n-1$. For $d \in \mathbb{N}$, denote $\mathcal{P}_d$ the set of $\mathbb{F}_2$-polynomials of degree at most $d-1$. We show that $\mathtt{BCH}(\mathcal{P}_{n/4})$ is the desired Sidon set.

▶ **Theorem 19.** *Let $n = 4d$ for some $d \in \mathbb{N}$, and $S = \mathtt{BCH}(\mathcal{P}_d) \subseteq \mathcal{P}_d \times \mathcal{P}_{3d} \subseteq \mathbb{F}_{2^d} \times \mathbb{F}_{2^{3d}} \cong \mathbb{Z}_2^n$. The Boolean function $\mathbf{1}_S : \mathbb{F}_2^n \to \{0,1\}$ satisfies that $\mathsf{RPDT}_{\mathrm{depth}}(\mathbf{1}_S) = O(\log n)$ and $\|\widehat{\mathbf{1}_S}\|_1 = \Omega(2^{n/8})$.*

Proof. Note that S is a Sidon set of size $|\mathcal{P}_d| = 2^d = 2^{n/4}$. The spectral norm lower bound follows from Proposition 17. It remains to upper-bound the randomized parity decision tree complexity.

We first describe a randomized algorithm that computes $\mathbf{1}_S$. Suppose a pair $(u, v) \in \mathbb{F}_2[x] \times \mathbb{F}_2[x]$ with $\deg(u) < d$ and $\deg(v) < 3d$ is a given. To determine whether $(u, v) \in S$, we interpret $v - u^3$ as a polynomial in $\mathbb{F}_{2^m}[x]$, where $m = \lceil \log(3d) \rceil + 2$, and check whether $v - u^3 \equiv 0$. This can be accomplished using the standard randomized polynomial identity

[3] For clarity, in the rest of this section, we use the notation $\mathbb{Z}_2$ to refer to the additive group of size 2.

testing algorithm: pick a random $t \in \mathbb{F}_{2^m}$ and evaluate $v(t) - u(t)^3 \in \mathbb{F}_{2^m}$. The algorithm declares that "$v \neq u^3$" if any of the evaluated values is non-zero, and declares "$v = u^3$" otherwise.

Notice that this randomized algorithm has a one-sided error, as it always makes the correct declaration when $v = u^3$ (i.e. $\mathbf{1}_S(u, v) = 1$). In the case of $v \neq u^3$, since the polynomial $v - u^3$ has at most $3d$ roots, the probability of error is at most $3d/2^m \leq 1/4$.

Next, we convert the randomized algorithm into a randomized parity decision tree. For every $t \in \mathbb{F}_{2^m}$, the set

$$H_t \coloneqq \{u \in \mathcal{P}_d : u(t) = 0\} \subseteq \mathcal{P}_d \cong \mathbb{Z}_2^d$$

is a linear subspace of co-dimension m in $\mathbb{Z}_2^d$, and each coset of H_t takes a fixed value when evaluated at t. Therefore, the value of $u(t)$ is determined by m deterministic linear queries to u. Similarly, the value of $v(t)$ is determined by m deterministic linear queries to v. Hence, we construct the randomized parity decision tree as follows: pick a random $t \in \mathbb{F}_{2^m}$ and make the $2m$ parity queries that determine the values of $u(t), v(t) \in \mathbb{F}_{2^m}$, and the decision tree outputs whether $v(t)$ and $u(t)^3$ are equal. ◀

5.1 Payley-Zygmud inequality and typical Fourier coefficients

Here instead of Hölder's inequality, we use the Payley-Zygmund inequality to deduce a stronger conclusion than Proposition 17 which could be of independent interest. We show that for the indicator of a Sidon set, a constant fraction of the Fourier coefficients is large. We have shown that for a Sidon set S in an Abelian group G,

$$\sum_{\chi \in \widehat{G}} |\widehat{\mathbf{1}_S}(\chi)|^2 = \frac{|S|}{|G|} \quad \text{and} \quad \sum_{\chi \in \widehat{G}} |\widehat{\mathbf{1}_S}(\chi)|^4 \leq \frac{2|S|^2}{|G|^3}.$$

Consider the random variable $X = |\widehat{\mathbf{1}_S}(\chi)|^2$ where χ is a character picked uniformly at random. The above bounds translate to $\mathbb{E}[X] = |S|/|G|^2$ and $\mathbb{E}[X^2] = 2|S|^2/|G|^4 = 2\,\mathbb{E}[X]^2$. By Paley-Zygmund inequality, we have for any $\delta \in (0, 1)$,

$$\Pr[X > \delta\,\mathbb{E}[X]] \geq (1 - \delta)^2 \frac{(\mathbb{E}[X])^2}{\mathbb{E}[X^2]} \geq \frac{(1 - \delta)^2}{2}.$$

Taking $\delta = 1/2$ for example, this gives

$$\Pr_{\chi}\left[|\widehat{\mathbf{1}_S}(\chi)| > \frac{\sqrt{|S|}}{\sqrt{2}|G|}\right] \geq \frac{1}{8}.$$

References

1 Nikhil Bansal and Makrand Sinha. k-forrelation optimally separates quantum and classical query complexity. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 1303–1316, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3406325.3451040.

2 R.C. Bose and D.K. Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and Control*, 3(1):68–79, March 1960. doi:10.1016/s0019-9958(60)90287-4.

3 Arkadev Chattopadhyay, Yogesh Dahiya, Nikhil S. Mande, Jaikumar Radhakrishnan, and Swagato Sanyal. Randomized versus deterministic decision tree size. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC '23. ACM, June 2023. doi:10.1145/3564246.3585199.

4 Arkadev Chattopadhyay, Shachar Lovett, and Marc Vinyals. Equality alone does not simulate randomness. In *34th Computational Complexity Conference (CCC 2019)*, 2019. doi:10.4230/LIPICS.CCC.2019.14.

5 Arkadev Chattopadhyay, Nikhil S. Mande, and Suhail Sherif. The log-approximate-rank conjecture is false. *J. ACM*, 67(4):23:1–23:28, 2020. doi:10.1145/3396695.

6 Bernard Chazelle. *The discrepancy method: randomness and complexity*. Cambridge University Press, 2001.

7 Bernard Chazelle and Alexey Lvov. A trace bound for the hereditary discrepancy. In *Proceedings of the sixteenth annual symposium on Computational geometry*, pages 64–69, 2000. doi:10.1145/336154.336179.

8 Tsun-Ming Cheung, Hamed Hatami, Kaave Hosseini, and Morgan Shirley. Separation of the factorization norm and randomized communication complexity. In *38th Computational Complexity Conference (CCC 2023)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.CCC.2023.1.

9 Tsun-Ming Cheung, Hamed Hatami, Rosie Zhao, and Itai Zilberstein. Boolean functions with small approximate spectral norm. *Electron. Colloquium Comput. Complex.*, TR22-041, 2022. URL: https://eccc.weizmann.ac.il/report/2022/041, arXiv:TR22-041.

10 Kenneth R. Davidson and Allan P. Donsig. Norms of schur multipliers. *Illinois Journal of Mathematics*, 51(3), July 2007. doi:10.1215/ijm/1258131101.

11 Uma Girish, Avishay Tal, and Kewen Wu. Fourier Growth of Parity Decision Trees. In *36th Computational Complexity Conference (CCC 2021)*, volume 200 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 39:1–39:36, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2021.39.

12 Mika Göös, Toniann Pitassi, and Thomas Watson. The landscape of communication complexity classes. *computational complexity*, 27(2):245–304, June 2018. doi:10.1007/s00037-018-0166-6.

13 Ben Green and Tom Sanders. Boolean functions with small spectral norm. *Geometric and Functional Analysis*, 18(1):144–162, March 2008. doi:10.1007/s00039-008-0654-y.

14 Lianna Hambardzumyan, Hamed Hatami, and Pooya Hatami. Dimension-free bounds and structural results in communication complexity. *Israel Journal of Mathematics*, 253(2):555–616, 2023.

15 G H Hardy, J E Littlewood, and Georg Polya. *Cambridge mathematical library: Inequalities*. Cambridge University Press, Cambridge, England, February 1988.

16 Alexis Hocquenghem. Codes correcteurs d'erreurs. *Chiffres*, 2:147–156, 1959.

17 Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. In *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*, STOC '91, pages 455–464, New York, NY, USA, 1991. Association for Computing Machinery. doi:10.1145/103418.103466.

18 Noam Nisan. Crew prams and decision trees. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 327–335, 1989. doi:10.1145/73007.73038.

19 Toniann Pitassi, Morgan Shirley, and Adi Shraibman. The strength of equality oracles in communication. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 89:1–89:19, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2023.89.

20 Tom Sanders. Boolean functions with small spectral norm, revisited. *Math. Proc. Camb. Philos. Soc.*, 167(02):335–344, September 2019.

21 Amir Shpilka, Avishay Tal, and Ben lee Volk. On the structure of boolean functions with small spectral norm. *Computational Complexity*, 26(1):229–273, September 2017. doi:10.1007/s00037-015-0110-y.

22 Avishay Tal. Tight Bounds on the Fourier Spectrum of AC0. In *32nd Computational Complexity Conference (CCC 2017)*, volume 79 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:31, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2017.15.

23 Hing Yin Tsang, Chung Hoi Wong, Ning Xie, and Shengyu Zhang. Fourier sparsity, spectral norm, and the log-rank conjecture. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 658–667, 2013. doi:10.1109/FOCS.2013.76.

Backdoor Defense, Learnability and Obfuscation

Paul Christiano[1]
Alignment Research Center, Berkeley, CA, USA

Jacob Hilton[2]
Alignment Research Center, Berkeley, CA, USA

Victor Lecomte
Alignment Research Center, Berkeley, CA, USA

Mark Xu[3]
Alignment Research Center, Berkeley, CA, USA

Abstract

We introduce a formal notion of defendability against backdoors using a game between an attacker and a defender. In this game, the attacker modifies a function to behave differently on a particular input known as the "trigger", while behaving the same almost everywhere else. The defender then attempts to detect the trigger at evaluation time. If the defender succeeds with high enough probability, then the function class is said to be defendable. The key constraint on the attacker that makes defense possible is that the attacker's strategy must work for a randomly-chosen trigger.

Our definition is simple and does not explicitly mention learning, yet we demonstrate that it is closely connected to learnability. In the computationally unbounded setting, we use a voting algorithm of Hanneke et al. [13] to show that defendability is essentially determined by the VC dimension of the function class, in much the same way as PAC learnability. In the computationally bounded setting, we use a similar argument to show that efficient PAC learnability implies efficient defendability, but not conversely. On the other hand, we use indistinguishability obfuscation to show that the class of polynomial size circuits is not efficiently defendable. Finally, we present polynomial size decision trees as a natural example for which defense is strictly easier than learning. Thus, we identify efficient defendability as a notable intermediate concept in between efficient learnability and obfuscation.

2012 ACM Subject Classification Theory of computation → Machine learning theory; Computing methodologies → Machine learning; Security and privacy → Mathematical foundations of cryptography; Theory of computation → Cryptographic primitives

Keywords and phrases backdoors, machine learning, PAC learning, indistinguishability obfuscation

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.38

Related Version *Version including Appendix A in full*: `https://arxiv.org/abs/2409.03077`

Acknowledgements We are grateful to Dmitry Vaintrob for an earlier version of the results in Appendix A; to Thomas Read for finding the "Backdoor with likely input patterns" example and for help with proofs; to Andrea Lincoln, Dávid Matolcsi, Eric Neyman, George Robinson and Jack Smith for contributions to the project in its early stages; and to Geoffrey Irving, Robert Lasenby and Eric Neyman for helpful comments on drafts.

[1] Work done while at the Alignment Research Center prior to April 2024.
[2] Corresponding author.
[3] Authors ordered alphabetically.


© Paul Christiano, Jacob Hilton, Victor Lecomte, and Mark Xu;
licensed under Creative Commons License CC-BY 4.0



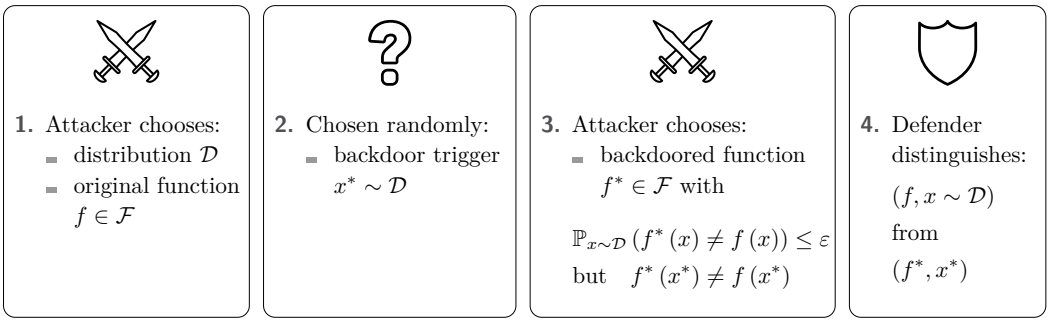


Figure 1 The game used to define ε-defendability for a class $\mathcal{F}$ of $0,1$-valued functions.

1 Introduction

A *backdoor* in a machine learning model is a modification to the model that causes it to behave differently on certain inputs that activate a secret "trigger". There is a wide literature on backdoor attacks and defenses from both a theoretical and an empirical perspective [26]. However, prior theoretical work typically makes reference to a particular training dataset, leading to a focus on data poisoning attacks. In this work we introduce a formal notion of a backdoor that allows the attacker to modify the model arbitrarily. Our definition is simple, but nevertheless gives rise to a rich array of strategies incorporating both learning and obfuscation.

Our formal notion is focused on backdoor detection *at runtime*, meaning that the defender is given a particular input and must flag whether or not it activates the backdoor trigger. We focus on this case because of the existence of *undetectable* backdoors if the defender is instead required to flag a model as backdoored without being given any particular input [11]. Moreover, detection at runtime is sufficient in threat models where the attacker is given only one opportunity to modify the model, being akin to giving the defender an additional chance to modify the model after the attacker. We also focus on the white-box setting, in which the defender has access to a complete description of the model.

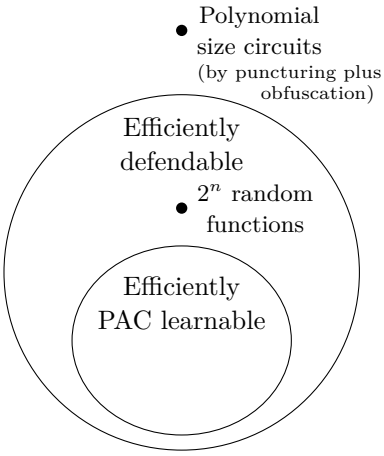
Our main contributions, which are also summarized in Figures 1 and 2, are as follows:

- **Definition of ε-defendability (Section 3).** We introduce a simple game between an attacker and a defender to define what it means for a representation class to be *ε-defendable* with a certain confidence. The only constraint placed on the attacker is that their backdoor strategy must work for a randomly-chosen trigger. Without this assumption, there is a symmetry between the original and backdoored functions that makes defense impossible.
- **Statistical defendability (Section 4).** In the absence of computational constraints on the defender, we show that a representation class $\mathcal{F}$ is ε-defendable with confidence tending to 1 if and only if $\varepsilon = o(1/\mathrm{VC}(\mathcal{F}))$, where $\mathrm{VC}(\mathcal{F})$ is the Vapnik–Chervonenkis dimension of $\mathcal{F}$. We achieve this by adapting a result of Hanneke et al. [13] that applies a learning algorithm multiple times and takes a majority vote.
- **Computational defendability (Section 5).** We introduce a notion of *efficient defendability* in which the defender's detection strategy must run in polynomial time. We show that efficient PAC learnability implies efficient defendability, but not conversely. We also show that under certain cryptographic assumptions, the class of polynomial size circuits is not efficiently defendable, by combining a puncturable pseudorandom function with an efficient indistinguishability obfuscator.

Statistical defendability (Section 4)

$\mathcal{F}$ is ε-defendable with confidence $\to 1$
$\iff$
$\varepsilon = o\left(1/\mathrm{VC}(\mathcal{F})\right)$
(by learning plus voting)

Computational defendability (Section 5)



Defendability of decision trees (Section 6)

Polynomial size decision trees are efficiently uniform-defendable in a single evaluation

(a) Statistical defendability is concerned with computationally unbounded defenders, and is essentially determined by the VC dimension, in much the same way as statistical learnability.

(b) Computational defendability is concerned with polynomial-time defenders, and the set of efficiently defendable classes lies strictly between the set of efficiently learnable classes and the set of all classes.

(c) In the setting where the distribution over inputs is uniform, polynomial size decision trees are a natural class for which defense is faster than learning.

Figure 2 Summary of the results in this paper.

- **Defendability of decision trees (Section 6).** In the setting where the distribution over inputs is uniform, we give a defense for polynomial size decision trees that runs in the time taken to evaluate a single decision tree. This provides a natural example for which defense is faster than learning.

We conclude in Section 7 with a discussion of whether efficient defendability can be further separated from efficient PAC learnability, so-called "mechanistic" defenses that resemble our defense for polynomial size decision trees, and the implications of our results for AI alignment.

2 Related work

The most closely related work to our own is that of Goldwasser, Kim, Vaikuntanathan, and Zamir [11], whose main results use a digital signature scheme to insert an undetectable backdoor into a model. The backdoor trigger can be applied to any input by perturbing it appropriately. In the black-box setting, the backdoor is undetectable in the sense that it is computationally infeasible for the defender to find any input on which the backdoored model and the original model differ. In the white-box setting, the backdoor is undetectable in the sense that it is computationally infeasible for the defender to distinguish the backdoored model from the original model. However, both of these results are in the setting where the backdoor trigger must be extracted from the model rather than detected at runtime.

In the defense at runtime setting, the same authors also show that a backdoored model can be modified by the defender to produce a model that is not backdoored. Once again, the backdoor trigger is applied by perturbing the input. The result only applies to models that

satisfy a certain smoothness assumption, and mirrors the use of randomized smoothing to achieve certified adversarial robustness by Cohen et al. [6]. By contrast, we avoid making smoothness assumptions, and further develop the theory of runtime backdoor detection.

Concurrent work by Goldwasser, Shafer, Vafa, and Vaikuntanathan [12] studies a black-box, non-realizable version of backdoor defense at runtime. Rather than predicting the exact label of the non-backdoored model on the given input (which is equivalent to our notion of runtime detection), the defender must instead produce a "canonical" label that has high accuracy on average, by using black-box access to the potentially backdoored model. In this local mitigation setting, they construct efficient defenders for almost-linear and almost-polynomial functions. They also consider the global mitigation setting, in which the defender must produce an entire clean model, constructing efficient global mitigators for Fourier-heavy functions.

Adversarial examples can be thought of as backdoor triggers, but the attacker must find similar inputs rather than a model with similar outputs. The gap between statistical and computational hardness has been explored in this context by Bubeck et al. [5] and Garg et al. [8]. Much like us, they find that defense is statistically possible but computationally hard.

Returning to backdoors, another thematically similar work to our own is that of Khaddaj et al. [23], which also questions the assumptions behind different backdoor attacks and defenses. However, it remains within the data-poisoning setting, in which the backdoor is inserted using corrupted training data.

A line of work beginning with Dumford and Scheirer [7] goes beyond data-poisoning, inserting a backdoor into an already-trained model, but still using data to search for the backdoored model. More manual methods of backdoor insertion were later introduced by Hong et al. [15]. This line of work is primarily empirical rather than theoretical.

Our result on statistical defendability is heavily based on the result of Hanneke et al. [13] in the realizable case. Following Levine and Feizi [25] and Jia et al. [18], they use a voting algorithm to produce a backdoor defense in the data-poisoning setting. By combining this with a classical result of Haussler, Littlestone, and Warmuth [14] on optimal learning, they relate the optimal performance of this algorithm to the VC dimension of the function class. Our result adapts this to our more general notion of defendability.

Our result on computational defendability of polynomial size circuits uses a hybrid argument that combines a puncturable pseudorandom function and an efficient indistinguishability obfuscator. This combination has been used in a number of previous hybrid arguments, as described by Sahai and Waters [29]. More generally, the relationship between learnability and cryptographic assumptions has a long history, with some basic constructions in this vein described in the textbook of Kearns and Vazirani [22].

3 Definition of ε-defendability

Our formal notion of defendability against backdoors is based on a game played between two players, an "attacker" and a "defender". Informally, the attacker starts with a function and is given a randomly sampled backdoor trigger. They must choose a backdoored function that almost always agrees with the original function, but disagrees with the original function on the backdoor trigger. Then the defender is given either the original function and a random input, or the backdoored function and the backdoor trigger, and they must distinguish one from the other with high confidence.

For our formal definition, we fix a security parameter $n \in \mathbb{N}$ and consider classifiers $\mathcal{X} \to \{0,1\}$ where $\mathcal{X} = \{0,1\}^n$. Formally, a *representation class* over $\mathcal{X}$ is a function $\{0,1\}^{R(n)} \to \{\text{all functions } \mathcal{X} \to \{0,1\}\}$ for some $R(n) \in \mathbb{N}$, but we generally keep the

choice of representation implicit and conflate a representation class with the set of functions in its image. In our results on computationally constrained defenders, we also implicitly assume that $R(n)$ is polynomial in n.

Our definition also makes use of the *example oracle* $\mathrm{Ex}(f, \mathcal{D})$ where $\mathcal{D}$ is a distribution over $\mathcal{X}$ and $f : \mathcal{X} \to \{0,1\}$. This is an oracle that outputs $(x, f(x))$ for a randomly sampled $x \sim \mathcal{D}$.

▶ **Definition 3.1.** Let $\mathcal{F}$ be a representation class over $\mathcal{X} = \{0,1\}^n$ and let $\varepsilon, \delta > 0$. We say that $\mathcal{F}$ is *ε-defendable with confidence* $1 - \delta$ if there is a probabilistic algorithm (the "detection strategy") that wins the following game with probability at least $1 - \delta$:

- An adversary with knowledge of the detection strategy (the "attacker") chooses a distribution $\mathcal{D}$ over $\mathcal{X}$ and $f \in \mathcal{F}$.
- We randomly sample $x^* \sim \mathcal{D}$ (the "backdoor trigger"), and then the adversary chooses $f^* \in \mathcal{F}$ (the "backdoored function").
- The backdoored function f^* is called *ε-valid* if

$$\mathbb{P}_{x \sim \mathcal{D}}(f^*(x) \neq f(x)) \leq \varepsilon \quad \text{but} \quad f^*(x^*) \neq f(x^*).$$

- We randomly choose (f', x') to pass to the detection strategy to be either (f, x) for a random sample $x \sim \mathcal{D}$, or (f^*, x^*), with 50% probability each. The detection strategy is given access to the example oracle $\mathrm{Ex}(f', \mathcal{D})$ as well as the representation of f', x', $f'(x')$, ε and δ as input, and must output either Acc (for "accept") or Rej (for "reject").
- If (f, x) is passed to the detection strategy, then the detection strategy wins if it outputs Acc. If (f^*, x^*) is passed to the detection strategy, then the detection strategy wins if either it outputs Rej or f^* is not ε-valid.

In this definition, the adversary is not specified by a collection of algorithms, but is instead shorthand for a universal quantifier. Thus we could have equivalently (but perhaps convolutedly) said, "there exists a detection strategy such that, for all distributions $\mathcal{D}$ over $\mathcal{X}$ and all $f \in \mathcal{F}$, the expectation over $x^* \sim \mathcal{D}$ of the minimum over all $f^* \in \mathcal{F}$ of the win probability of the detection strategy is at least $1 - \delta$".

The manner in which f, x^* and f^* are chosen in this definition deserves some elaboration. If the attacker were allowed to choose the backdoor trigger x^* themselves, then it would be too easy for the adversary to insert an undetectable backdoor. So we instead require the adversary to come up with a backdoor insertion strategy that works for a randomly-chosen trigger. In fact, we do not even allow the adversary to see the backdoor trigger when they are selecting the original function f, or else they could produce f^* by *removing an existing backdoor* rather than *inserting a new one*. This is illustrated by the following example.

▶ **Example** (Existing backdoor removal). Let $\mathcal{X} = \{0,1\}^n$ and let $\mathcal{F} = \{\mathbb{1}_x \mid x \in \mathcal{X}\}$, where $\mathbb{1}_x : \mathcal{X} \to \{0,1\}$ denotes the indicator function

$$\mathbb{1}_x(x') = \begin{cases} 1, & \text{if } x' = x \\ 0 & \text{if } x' \neq x. \end{cases}$$

Take $\mathcal{D}$ to be uniform over $\mathcal{X}$. If we allowed the adversary to choose both f and f^* after sampling $x^* \sim \mathcal{D}$, then they could take $f = \mathbb{1}_{x^*}$ and $f^* = \mathbb{1}_{x'}$ for $x' \sim \mathcal{D}$ (or in the event that $x' = x^*$, instead take $f = \mathbb{1}_{x''}$ for some $x'' \in \mathcal{X} \setminus \{x^*\}$ sampled uniformly at random). Then f^* would be $\frac{2}{2^n}$-valid, and (f, x) and (f^*, x^*) would be identically distributed for $x \sim \mathcal{D}$. So if $\varepsilon \geq \frac{1}{2^{n-1}}$ then any detection strategy would win with probability at most $\frac{1}{2}$. This would be quite an extreme failure for the defender, especially considering that the VC dimension of $\mathcal{F}$ is only 1.

Thus the way in which f, x^* and f^* are chosen, by randomly sampling the backdoor trigger x^* after the original function f is chosen but before the backdoored function f^* is chosen, is the key symmetry-breaking assumption that makes the game interesting.

▶ Remark. We could generalize this definition to include functions $\mathcal{X} \to \mathcal{Y}$ where $\mathcal{Y}$ is any subset of $\mathbb{R}$ (or even more generally, any metric space). In this setting, the most natural generalization of ε-validity is

$$\mathbb{E}_{x\sim\mathcal{D}}\left[\left(f^*(x) - f(x)\right)^2\right] \leq \varepsilon \quad \text{but} \quad \left|f^*(x^*) - f(x^*)\right| \geq 1.$$

4 Statistical defendability

In this section we will completely determine, up to a constant factor, the values of ε and δ for which an arbitrary representation class is ε-defendable with confidence $1-\delta$, in the absence of any computational constraints on the detection strategy.

Our result is expressed in terms of the *Vapnik–Chervonenkis (VC) dimension* of the representation class $\mathcal{F}$, denoted $\mathrm{VC}(\mathcal{F})$. This is defined as the maximum size of a set S of points *shattered* by $\mathcal{F}$, meaning that all $2^{|S|}$ functions $S \to \{0,1\}$ are realized as the restriction $f|_S$ of some $f \in \mathcal{F}$.

▶ **Theorem 4.1.** *Let $\mathcal{F}$ be a representation class over $\{0,1\}^n$ and let $\varepsilon > 0$. Then the most (i.e., supremum) confidence with which $\mathcal{F}$ is ε-defendable is*

$$\max\left(\tfrac{1}{2}, 1 - \Theta\left(\mathrm{VC}(\mathcal{F})\,\varepsilon\right)\right)$$

as $\mathrm{VC}(\mathcal{F}) \to \infty$.

In particular, $\mathcal{F}$ is ε-defendable with confidence tending to 1 as $n \to \infty$ if and only if $\varepsilon = o\left(1/\mathrm{VC}(\mathcal{F})\right)$ as $n \to \infty$.

Thus in the computationally unbounded setting, defendability is determined almost entirely by the VC dimension of the representation class, in much the same way as PAC learnability [2, 22, Chapter 3].

In fact, to prove Theorem 4.1, we will make use of a certain kind of learning algorithm called a *prediction strategy*. Intuitively, a prediction strategy uses the value of a function on random samples to predict the value of the function on a new random sample. The formal definition is as follows.

▶ **Definition.** Let $\mathcal{F}$ be a representation class over $\mathcal{X} = \{0,1\}^n$. A *prediction strategy* for $\mathcal{F}$ is a randomized algorithm that is given access to the example oracle $\mathrm{Ex}(f, \mathcal{D})$ for some distribution $\mathcal{D}$ over $\mathcal{X}$ and some $f \in \mathcal{F}$, as well as $x \in \mathcal{X}$ as input, and outputs a prediction for $f(x)$.

The *sample size* of a prediction strategy is the maximum number of times it calls the example oracle, maximizing over any randomness used by the algorithm, any calls to the oracle, and the choice of $\mathcal{D}$, f and x.

Given a choice of $\mathcal{D}$ and f, the *error rate* of a prediction strategy is the probability that it fails to correctly predict $f(x)$ for $x \sim \mathcal{D}$, randomizing over any randomness used by the algorithm, any calls to the oracle, and the choice of x.

In their classical paper on prediction strategies, Haussler, Littlestone, and Warmuth [14] exhibit a "1-inclusion graph" algorithm yielding the following result.

▶ **Theorem** (Haussler–Littlestone–Warmuth). *For any representation class $\mathcal{F}$ and any positive integer m, there is a prediction strategy for $\mathcal{F}$ with sample size $m-1$ such that for any distribution $\mathcal{D}$ over $\mathcal{X}$ and any $f \in \mathcal{F}$, the error rate is at most*

$$\frac{\mathrm{VC}(\mathcal{F})}{m}.$$

As discussed in that work, there is a close relationship between prediction strategies and PAC learning algorithms, which we define in Section 5.2. We focus on prediction strategies in this section because they make our proof easier to carry out while avoiding the introduction of unnecessary logarithmic factors.

Our proof of Theorem 4.1 is closely modeled on a proof of Hanneke et al. [13, Theorem 3.1], although we cannot quote that result directly since it is specialized to a data poisoning setup. As in that proof, our detection strategy works by applying the Haussler–Littlestone–Warmuth prediction strategy multiple times and taking a majority vote.

The basic idea for the detection strategy is that, given (f', x') as input (which could be either (f, x) or (f^*, x^*)), if we could successfully predict $f(x')$, then we could distinguish the two cases by comparing this to $f'(x')$, since $f(x) = f(x)$ but $f(x^*) \neq f^*(x^*)$ if f^* is ε-valid. To attempt to make this prediction, we use the Haussler–Littlestone–Warmuth prediction strategy. However, if $f' = f^*$ then this can fail if we encounter a point where f and f^* disagree. Hence we are forced to use a small sample size, but we are still left with a small constant probability of failure. Fortunately though, we can repeat this procedure multiple times with independent samples and take a majority vote, which allows us to make this small constant probability vanish.

We now provide a sketch proof of Theorem 4.1. For a full proof, see Appendix B.

Sketch proof of Theorem 4.1. Write $d = \mathrm{VC}(\mathcal{F})$, and note that the detection strategy can achieve a win probability of at least $\frac{1}{2}$ simply by guessing randomly.

To show that the most confidence with which $\mathcal{F}$ is ε-defendable is at most $\max\left(\frac{1}{2}, 1 - \Omega(d\varepsilon)\right)$, let S be a set of size d shattered by $\mathcal{F}$. Roughly speaking, the adversary can take $\mathcal{D}$ to be uniform over S, and take f to be uniform over the witnesses to the shattering. Then the adversary can insert a backdoor by simply changing the value of f at x^* and no other points of S. The actual construction is slightly more careful than this, and is given in the full proof.

To show that the most confidence with which $\mathcal{F}$ is ε-defendable is at least $\max\left(\frac{1}{2}, 1 - O(d\varepsilon)\right)$, our detection strategy is as follows. Given (f', x') as input, we apply the Haussler–Littlestone–Warmuth prediction strategy with sample size $m-1$ and error rate at most $\frac{d}{m}$ to make a prediction for $f'(x')$, which we in turn think of as a prediction for $f(x')$. We repeat this procedure r times with independent samples and take a majority vote: if more than half of the predictions are different from $f'(x')$, then we output Rej, and otherwise we output Acc.

If (f, x) is passed to the detection strategy, then the probability that more than $\frac{1}{2}$ of the votes are wrong is at most $\frac{2d}{m}$, by Markov's inequality. If (f^*, x^*) is passed to the detection strategy, then there are two ways in which each vote can be wrong: either f and f^* disagree on at least one of the $m-1$ samples, or the prediction strategy fails to predict $f(x^*)$ even if it is provided with the value of f on all of these $m-1$ samples. The probability that $\frac{1}{4}$ or more of the votes will be wrong due to the second failure mode is at most $\frac{4d}{m}$, by Markov's inequality again. Meanwhile, we can ensure that the probability of the first failure mode for each vote is at most $\frac{1}{5}$ by taking $\frac{1}{5\varepsilon} < m \leq \frac{1}{5\varepsilon} + 1$. Since each choice of $m-1$ samples was independent, the probability that $\frac{1}{4}$ or more of the votes will be wrong due to the first

failure mode is at most $\exp\left(-\frac{r}{200}\right)$, by Hoeffding's inequality, which can be made arbitrarily small by taking r to be sufficiently large. Hence the probability that $\frac{1}{2}$ or more of the votes will be wrong is at most $\frac{4d}{m}$, and so the overall win probability is $1 - \frac{3d}{m} = 1 - O(\varepsilon d)$, as required. ◀

Intuitively, the reason that this detection strategy works is that the if ε is small compared to $1/\mathrm{VC}(\mathcal{F})$, then the attacker must choose a backdoored function that is very "strange". Hence the detection strategy can sample a new function that is similar to the given possibly backdoored function, but more "normal", and see if the two functions agree on the given possible backdoor trigger. This process can be thought of as a kind of regularization.

While our proof used distillation (i.e., learning from function samples) plus ensembling (i.e., voting), other methods of regularization may also work. For example, we could also have sampled from a Boltzmann distribution centered on the given function (with exponentially decaying probability based on Hamming distance), to get the same result up to logarithmic factors. We prove this claim in Appendix A.

5 Computational defendability

5.1 Definition of efficient defendability

In Section 4, we related the defendability of a representation class to its VC dimension. However, the prediction strategy we used to construct the detection strategy runs in exponential time, since it involves an expensive search over orientations of a certain graph. Hence it is interesting to ask what a polynomial-time detection strategy can achieve. To explore this, we introduce the following notion.

▶ **Definition 5.1.** Let $\mathcal{F}$ be a representation class over $\{0,1\}^n$. We say that $\mathcal{F}$ is *efficiently defendable* if there is some polynomial p such that for any $\delta > 0$ and any $\varepsilon > 0$ with $\varepsilon < 1/p\left(n, \frac{1}{\delta}\right)$, $\mathcal{F}$ is ε-defendable with confidence $1 - \delta$ using a detection strategy that runs in time polynomial in n and $\frac{1}{\delta}$.

Note that the condition that $\varepsilon < 1/p\left(n, \frac{1}{\delta}\right)$ is crucial: if $\mathcal{F}$ were required to be ε-defendable with confidence $1 - \delta$ for any $\varepsilon, \delta > 0$, then this would not be possible even for a detection strategy with unlimited time (except in trivial cases), by Theorem 4.1. (Recall that ε constrains the attacker, which is why this requirement helps the defender.)

As a reminder, we also assume in this section that representation classes have polynomial-length representations.

5.2 Efficient defendability and efficient PAC learnability

In this sub-section we show that efficient PAC learnability implies efficient defendability, but not conversely.

The well-studied model of probably approximately correct (PAC) learning was introduced by Valiant [30]. The notion of efficient PAC learnability bears some resemblance to our notion of efficient defendability. The following definition is taken from Kearns and Vazirani [22, Definition 4].

▶ **Definition.** Let $\mathcal{F}$ be a representation class over $\mathcal{X} = \{0,1\}^n$. We say that $\mathcal{F}$ is *efficiently PAC learnable* if there is a probabilistic algorithm (the "PAC learning algorithm") with the following properties. The PAC learning algorithm is given access to the example oracle $\mathrm{EX}(f, \mathcal{D})$ for some distribution $\mathcal{D}$ over $\mathcal{X}$ and some $f \in \mathcal{F}$, as well as $0 < \tilde{\varepsilon} < \frac{1}{2}$ (the "error

parameter") and $0 < \tilde{\delta} < \frac{1}{2}$ (the "confidence parameter") as input. The PAC learning algorithm must run in time polynomial in n, $\frac{1}{\tilde{\varepsilon}}$ and $\frac{1}{\tilde{\delta}}$, and must output some polynomially evaluatable hypothesis h, i.e. a representation for a function $\mathcal{X} \to \{0,1\}$ that is evaluatable in time polynomial in n. The hypothesis must satisfy $\mathbb{P}_{x\sim\mathcal{D}}\left(f\left(x\right) \neq h\left(x\right)\right) \leq \tilde{\varepsilon}$ with probability at least $1 - \tilde{\delta}$ for any distribution $\mathcal{D}$ over $\mathcal{X}$ and any $f \in \mathcal{F}$.

Note that $\tilde{\varepsilon}$ and $\tilde{\delta}$ play subtly different roles in this definition to ε and δ in the definition of efficient defendability. In particular, there is no condition on $\tilde{\varepsilon}$ as a function of n and $\tilde{\delta}$ in the definition of efficient PAC learnability.

We now show that efficient PAC learnability implies efficient defendability. This follows easily from the proof of Theorem 4.1. Even though that result used a prediction strategy that runs in exponential time, it is straightforward to replace it by an efficient PAC learning algorithm.

▶ **Corollary 5.2.** *Let $\mathcal{F}$ be a representation class over $\{0,1\}^n$. If $\mathcal{F}$ is efficiently PAC learnable, then $\mathcal{F}$ is efficiently defendable.*

Proof. If $\mathcal{F}$ is efficiently PAC learnable, then for any sufficiently large positive integer m, there is a polynomial-time prediction strategy for $\mathcal{F}$ with sample size $m-1$ and error rate at most $\frac{1}{6}\delta$: we simply PAC learn a hypothesis with confidence parameter $\frac{1}{12}\delta$ and error parameter $\frac{1}{12}\delta$, and then evaluate this hypothesis on the given point. The definition of PAC learning guarantees that this succeeds as long as m exceeds a particular polynomial in n and $\frac{1}{\delta}$.

Our detection strategy is then exactly the same as the one from the proof of Theorem 4.1, but using the above prediction strategy instead of the Haussler–Littlestone–Warmuth prediction strategy. Since $\varepsilon < 1/p(n,\frac{1}{\delta})$ for some polynomial p of our choice, choosing $m > \frac{1}{5\varepsilon}$ as in that proof suffices to ensure that m is eventually large enough for the prediction strategy to succeed. Finally, we take the number of votes r to be the smallest integer greater than $200\log\left(\frac{1}{\delta}\right)$, which keeps the time complexity of the detection strategy polynomial in n and $\frac{1}{\delta}$. Then the detection strategy's overall win probability is at least $1 - \frac{3}{6}\delta - \frac{1}{2}\exp\left(-\frac{r}{200}\right) > 1-\delta$, as required. ◀

We now show that the converse implication does not hold in the random oracle model of computation. In this model, programs have access to an oracle that outputs the result of calling of a uniformly random function $\{0,1\}^* \to \{0,1\}$.

▶ **Theorem 5.3.** *In the random oracle model of computation, with probability $1 - O(2^{-n^c})$ for all c over the choice of random oracle, there is a polynomially evaluatable representation class over $\mathcal{X} = \{0,1\}^n$ that is efficiently defendable but not efficiently PAC learnable.*

To prove this, we simply take a representation class of 2^n functions that make unique calls to the random oracle (supplying a different input to the oracle whenever either the function is different or the input to the function is different), and show that, with high probability over the choice of random oracle, this representation class is efficiently defendable but not efficiently PAC learnable. Roughly speaking, it is efficiently defendable because, for most sets of 2^n random functions with 2^n possible inputs, every pair of functions differs on around half of inputs, meaning that there are no ε-valid functions for the adversary to choose from. Meanwhile, it is not efficiently PAC learnable because most random functions with 2^n possible inputs differ from every possible function that a polynomial-time algorithm could output on around half of inputs. For a full proof, see Appendix C.

Although this result is in the random oracle model of computation, the same proof shows that the converse does not hold in the usual model of computation either. This is because we can make the random choices "on the outside": we fix a randomly-chosen lookup

table, and have the functions use that instead of calling the random oracle. However, this representation class is not necessarily polynomially evaluatable, since the lookup table would take up exponential space.

Nevertheless, we conjecture that there is a polynomially evaluatable counterexample that uses a pseudorandom function instead of a random oracle. The above proof cannot be immediately adapted to work for an arbitrary pseudorandom function, because two distinct keys could give rise to very similar functions, but it might not be too challenging to adapt it.

▶ **Conjecture 5.4.** *Assuming* OWF*, there is a polynomially evaluatable representation class over* $\{0,1\}^n$ *that is efficiently defendable but not efficiently PAC learnable.*

Here, OWF denotes the existence of a one-way function, which can used to construct a pseudorandom function [9, 20, Chapter 6].

One way in which our counterexample is unsatisfying is that it relies on the adversary being unable to find a backdoored function that is ε-valid, making detection trivial. However, it is straightforward to extend this result to a class for which it is always possible for the adversary to find a backdoored function that is ε-valid, but that is nevertheless efficiently defendable.

▶ **Example** (Special-cased random functions)**.** Let $\widetilde{\mathcal{F}}$ be one of the above representation classes over $\mathcal{X} = \{0,1\}^n$ that serves as a counterexample (either 2^n functions that call a random oracle, or a random choice of 2^n functions). Now let

$$\mathcal{F} = \left\{ x \mapsto \begin{cases} f(x), & \text{if } x \neq x^* \\ y & \text{if } x = x^* \end{cases} \;\middle|\; f \in \widetilde{\mathcal{F}}, x^* \in \mathcal{X}, y \in \{0,1\} \right\},$$

where implicitly, the representation of a function in $\mathcal{F}$ is given by a triple consisting of the representation of the function $f \in \widetilde{\mathcal{F}}$, the special-cased point $x^* \in \mathcal{X}$, and the special-cased value $y \in \{0,1\}$. Then the adversary can ensure that there is always a $\frac{2}{2^n}$-valid backdoored function by taking $\mathcal{D}$ to be the uniform distribution over $\mathcal{X}$. However, $\mathcal{F}$ is still efficiently defendable with high probability, since the detection strategy can check whether or not the input has been special-cased. $\mathcal{F}$ also remains not efficiently PAC learnable with high probability.

Thus we have an example of a non-trivial detection strategy that is faster than any PAC learning algorithm for that representation class. This shows that the learning-based detection strategy of Corollary 5.2 is not always best possible under computational constraints. Nevertheless, our example is somewhat contrived, and so in Section 6 we give an example of a more natural representation class for which defense is faster than learning.

5.3 Efficient defendability and obfuscation

We conclude our analysis of computational defendability with the following result, which essentially says that representation classes that are rich enough to support obfuscation are not efficiently defendable.

▶ **Theorem 5.5.** *Assuming* OWF *and* iO*, the representation class* $\mathcal{F}$ *of polynomial size Boolean circuits over* $\mathcal{X} = \{0,1\}^n$ *is not efficiently defendable.*

Here, OWF again denotes the existence of a one-way function, and iO denotes the existence of an efficient indistinguishability obfuscator. Roughly speaking, an efficient indistinguishability obfuscator is a probabilistic polynomial-time algorithm that takes in

a circuit and outputs an "obfuscated" circuit with the same behavior. The circuit being "obfuscated" means that the obfuscations of two different circuits with the same behavior are not distinguishable by any probabilistic polynomial-time adversary. For a precise definition of an efficient indistinguishability obfuscator, we refer the reader to Sahai and Waters [29, Section 3]. The study indistinguishability obfuscation was initiated by Barak et al. [1], and an efficient indistinguishability obfuscator was constructed from well-studied computational hardness assumptions by Jain, Lin and Sahai [17].

We use OWF in our proof of Theorem 5.5 to obtain a puncturable pseudorandom function. Roughly speaking, a puncturable pseudorandom function is pseudorandom function such that any key (i.e., seed) can be "punctured" at a set of points. The key being "punctured" means that, when run using the punctured key, the pseudorandom function behaves the same on unpunctured points, but to a probabilistic polynomial-time adversary with knowledge of the punctured key only, the function looks pseudorandom on punctured points when run using the original key. For a precise definition of a puncturable pseudorandom function, we again refer the reader to Sahai and Waters [29, Section 3]. The observation that puncturable pseudorandom functions can be constructed from one-way functions was made by Boneh and Waters [3], Boyle et al. [4] and Kiayias et al. [24].

Our proof of Theorem 5.5 is an example of a *hybrid argument*: we show that two distributions are computationally indistinguishable via a sequence of intermediate distributions. For a precise definition of computational indistinguishability and further discussion of hybrid arguments, we refer the reader to Katz and Lindell [20, Chapter 6.8]. The specific combination of a puncturable pseudorandom function and an efficient indistinguishability obfuscator has been used in a number of previous hybrid arguments, as described by Sahai and Waters [29].

With these preliminaries in place, we are now ready to prove Theorem 5.5.

Proof of Theorem 5.5. Take $\varepsilon = 2^{-n}$. We will show that, for any polynomial p, the representation class $\mathcal{F}$ of polynomial size Boolean circuits is not ε-defendable with confidence $\frac{1}{2} + \frac{1}{p(n)}$ using a detection strategy that runs in time polynomial in n.

To see this, by OWF, let $C_K \in \mathcal{F}$ be the circuit for a puncturable pseudorandom function with key $K \in \{0,1\}^n$ and a single output bit, and by iO, let $i\mathcal{O}$ be an efficient indistinguishability obfuscator. The adversary proceeds by taking $\mathcal{D}$ to be uniform over $\mathcal{X}$, and takes $f = i\mathcal{O}(C_K)$ for some $K \in \{0,1\}^n$ chosen uniformly at random. As before, we may treat f as if it were chosen randomly, since the detection strategy's worst-case performance can be no better than its average-case performance. Finally, given x^*, the adversary takes

$$f^* = i\mathcal{O}\left(x \mapsto \begin{cases} C_{\langle \textsc{Punc}(K,x^*)\rangle}(x), & \text{if } x \neq x^* \\ \langle 1 - C_K(x^*)\rangle, & \text{if } x = x^* \end{cases}\right),$$

where $\textsc{Punc}(K, x^*)$ punctures the key K at the point x^*, and angle brackets $\langle \dots \rangle$ indicate values that are hard-coded into the circuit, rather than being computed by the circuit. Note that f^* is ε-valid since it agrees with f on every point except x^*.

It remains to show that no polynomial-time detection strategy can win against this adversary with probability at least $\frac{1}{2} + \frac{1}{p(n)}$, which is equivalent to saying that (f, x) and (f^*, x^*) are computationally indistinguishable for $x \sim \mathcal{D}$. To see this, consider the intermediate circuit

$$\tilde{f} = i\mathcal{O}\left(x \mapsto \begin{cases} C_{\langle \textsc{Punc}(K,x^*)\rangle}(x), & \text{if } x \neq x^* \\ \langle C_K(x^*)\rangle, & \text{if } x = x^* \end{cases}\right),$$

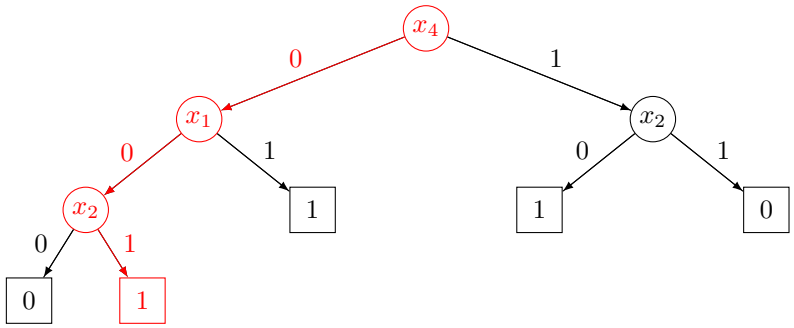


Figure 3 An example of a decision tree f over $\{0,1\}^4$ with a red path showing how $f(0110) = 1$.

which agrees with f everywhere. We claim that

$$(f, x) \overset{c}{\equiv} (f, x^*) \overset{c}{\equiv} (\tilde{f}, x^*) \overset{c}{\equiv} (f^*, x^*),$$

where $\overset{c}{\equiv}$ denotes computational indistinguishability of distributions. The first equivalence is simply an identity of distributions, since x^* was chosen randomly. The second equivalence follows by definition of indistinguishability obfuscation, with the technical detail that x^* is used as the "program state" that is passed from the circuit generator to the distinguisher. The third equivalence follows from puncturability: if $(\tilde{f}, x^*)$ and (f^*, x^*) were computationally distinguishable, then we could use this to computationally distinguish $C_K(x^*)$ from a uniformly random bit using only x^* and $\text{PUNC}(K, x^*)$. This final construction that relies on the fact that $i\mathcal{O}$ runs in polynomial time. By transitivity of computational indistinguishability, we have $(f, x) \overset{c}{\equiv} (f^*, x^*)$, as required. ◀

Thus even though all efficiently PAC learnable representation classes are efficiently defendable, some representation classes are not efficiently defendable due to the possibility of obfuscation.

6 Defendability of decision trees

In Section 5.2 we gave an example of a representation class for which defense is faster than learning, in the sense that it is efficiently defendable using a detection strategy that is faster than any PAC learning algorithm. However, our example was somewhat contrived, and was not polynomially evaluatable without access to a random oracle. In this section, we give an example of a more natural representation class for which defense is faster than learning: the class of polynomial size decision trees over $\{0,1\}^n$.

▶ **Definition.** A *decision tree* over $\{0,1\}^n$ is a rooted binary tree where each non-leaf node is labeled with one of the n input variables and each leaf node is labeled with a 0 or a 1. To evaluate a decision tree, we start at the root node and go left if the variable evaluates to 0 and right if it evaluates to 1, continuing until we reach a leaf. The *size* of a decision tree is its number of leaves.

An example of a decision tree and how to evaluate it is given in Figure 3.

In our study of decision trees, we focus on the *uniform-PAC* model of learning, in which the distribution $\mathcal{D}$ is always the uniform distribution. Thus we say that a representation class is *efficiently uniform-PAC learnable* to mean that it is efficiently PAC learnable as long as $\mathcal{D}$ is uniform. Similarly, we say that a representation class is *efficiently uniform-defendable* to mean that it is efficiently defendable as long as $\mathcal{D}$ is uniform.

As far as we are aware, it is unknown whether the representation class of polynomial size decision trees over $\{0,1\}^n$ is efficiently uniform-PAC learnable, although polynomial-time PAC learning algorithms are known for certain product distributions [19]. However, the class is efficiently uniform PAC-learnable if we allow membership queries (i.e., calls to an oracle that outputs the value of the function on an input of the algorithm's choice) [10, 27, Chapter 3.5]. By specializing the proof of Corollary 5.2, it follows that the representation class of polynomial size decision trees over $\{0,1\}^n$ is efficiently defendable, since the defender can efficiently compute the results of membership queries for themselves using the representation of the decision tree.

Nevertheless, these learning algorithms for decision trees generally use at least linearly many calls to the example or membership oracle. By contrast, we now show that there is a defense for decision trees that is much faster than this.

▶ **Theorem 6.1.** *The representation class $\mathcal{F}$ of decision trees over $\mathcal{X} = \{0,1\}^n$ of size at most s, where s is polynomial in n, is efficiently uniform-defendable using a detection strategy that makes 0 calls to the example oracle and runs in time O(time taken to evaluate a single decision tree).*

To prove this, we use a detection strategy that simply checks the depth of the leaf reached by the given input, and outputs Rej if this is too large. Roughly speaking, the reason this works is that, in order for the backdoored function to be ε-valid, the depth of the leaf reached by the backdoor trigger must be large for either the original function or the backdoored function. But since our trees have polynomially many leaves, this depth is unlikely to be large for the original function. Hence if the depth of the given function on the given input is large, then it is more likely that we have been given the backdoored function and the backdoor trigger.

For a full proof of Theorem 6.1, see Appendix D. We suspect that a similar approach would lead to fast detection strategies for other natural representation classes of piecewise constant functions, perhaps even in the non-uniform setting.

The detection strategy in this proof is not directly comparable to a learning algorithm, since it does not use the example oracle at all. However, if we treat a call to the example oracle as having similar computational cost to evaluating a single decision tree, then it is substantially faster than any possible learning algorithm for this representation class. Thus there is more to defense than only learning.

7 Discussion

7.1 Separating efficient defendability from efficient PAC learnability

A central theme of this work has been that learning can be used to perform backdoor defense, but backdoor defense cannot necessarily be used to perform learning. The first of these claims is encapsulated by Theorem 4.1 in the computationally unbounded setting and Corollary 5.2 in the computationally bounded setting. For the second of these claims, we have made several steps in this direction:

- We showed in Theorem 5.3 that efficient defendability does not imply efficient PAC learnability in the random oracle model of computation.
- We deduced from this that efficient defendability does not imply efficient PAC learnability in the usual model of computation either, as long as we allow representation classes that are not polynomially evaluatable, and we conjectured that there is a polynomially evaluatable counterexample assuming the existence of a one-way function.

- We showed in Theorem 6.1 that the representation class of polynomial size decision trees is efficiently uniform-defendable using a detection strategy that is faster than any possible learning algorithm.

Nevertheless, we still do not have an example of a natural representation class that is efficiently defendable but not efficiently PAC learnable. We consider finding such an example to be a central challenge for follow-up research. One possible candidate of independent interest is the representation class of shallow (i.e., logarithmic-depth) polynomial size Boolean circuits, or equivalently, polynomial size Boolean formulas.

▶ **Question 7.1.** Under reasonable computational hardness assumptions, is the representation class of polynomial size, logarithmic-depth Boolean circuits over $\{0,1\}^n$ efficiently defendable?

This representation class is known to not be efficiently PAC learnable under the assumption of the computational hardness of discrete cube roots [21, 22, Chapter 6].

7.2 Mechanistic defenses

Our detection strategy for decision trees in Theorem 6.1 is interesting not only because it is faster than any possible learning algorithm, but also because it works in a fundamentally different way. Given (f', x') as input, it does not run the decision tree f' on any inputs other than x' itself, but instead exploits the *mechanism* by which the value of $f'(x')$ is computed, by looking at the depth of the corresponding leaf. We call a defense that exploits the structure of f' in this kind of way *mechanistic*.

The existence of mechanistic defenses is another reason to suspect that there are other representation classes for which defense is strictly easier than learning. However, some sophistication may be required to construct such defenses. For example, one might hope to detect a backdoor trigger for a Boolean formula by checking the pattern of inputs to each gate, and seeing how unlikely that pattern would be for a random input to the formula. Unfortunately, though, the following example, found by Thomas Read, presents an obstacle to such an approach.

▶ **Example** (Backdoor with likely input patterns)**.** Consider the Boolean formulas $f, f^* : \{0,1\}^n \to \{0,1\}$ given by $f(x_1, \ldots, x_n) = x_n x_{n-1}$ and

$$f^*(x_1, \ldots, x_n) = (\ldots((x_1 \vee x_n x_2)\, x_2 \vee x_n x_3)\, x_3 \vee \cdots \vee x_n x_{n-1})\, x_{n-1},$$

where ab is shorthand for $(a \wedge b)$. By the distributive property, $f^*(x_1, \ldots, x_n)$ is logically equivalent to $x_1 x_2 \ldots x_{n-1} \vee x_n x_{n-1}$, and so f and f^* disagree only on the input $x^* := (1, \ldots, 1, 0)$. But on a uniformly random input to the formula, every gate in f^* receives every pattern of inputs with probability at least $1/8$. By contrast, in the logically equivalent version of f^*, the subformula $x_1 x_2 \ldots x_{n-1}$ is 1 on x^*, which only happens with probability $1/2^{n-1}$ on a uniformly random input.

This example suggests a possible mechanistic defense for Boolean formulas that involves first rewriting the formula. More generally, mechanistic defenses offer an exciting avenue for future research.

7.3 Implications for AI alignment

We are motivated to study backdoors as an analogy for *deceptive alignment*, the possibility that an advanced AI system would learn to behave cooperatively when there are subtle cues that it is in training, but uncooperatively when those cues are missing [16]. A deceptively

aligned model is analogous to a backdoored function in that it behaves similarly to a fully cooperative model except on certain inputs that are rare during training. Thus, in this analogy, $\mathcal{D}$ is the training distribution, f is the fully cooperative model, f^* is the deceptively aligned model, and x^* is an input on which f^* behaves uncooperatively.

Note that allowing the backdoor to be detected at runtime is appropriate in this analogy, because we have the opportunity to modify f^* or run a monitoring system. The flaw in the analogy comes from how the attacker is restricted. In our formal notion of defendability, the attacker must insert a backdoor that works for a randomly-chosen trigger. But in the case of deceptive alignment, there is no such restriction, and instead the backdoored behavior must arise from properties of the model architecture and the training distribution.

From this perspective, backdoor defenses that rely on learning are unsatisfying, because our assumption is that a similar process of learning gave rise to deceptive alignment in the first place. The detection strategy in our proof of Theorem 4.1 used distillation plus ensembling to produce a regularized model, but this may be exploiting the flaw in our analogy by using resampling to avoid the randomly-chosen trigger. Moreover, we would have to rely on a fast approximation to this strategy, since the exact version runs in exponential time. Similar issues apply to other methods of regularization, such as the one discussed in Appendix A, although that alternative does show more promise.

On the other hand, mechanistic defenses, as discussed in Section 7.2, may fare better, since they work very differently. Intuitively, if a detection strategy could spot the mechanism by which a deceptively aligned model concluded that it was in training, then it should transfer well to the case of deceptive alignment. A mechanistic defense may be able to do this by exploiting the fact that this mechanism is active on the backdoor trigger but inactive on most random inputs. Unfortunately though, we are lacking in examples of mechanistic defenses, which why we are excited to see more research in this direction.

Our result that polynomial size circuits are not efficiently defendable in Theorem 5.5 is also relevant to this analogy. Although it is unlikely that our indistinguishability obfuscator-based construction would arise out of ordinary model training, it is much more plausible for a trained model to be obfuscated in a more informal sense. Indeed, reverse engineering trained neural networks is an active area of research [28]. Hence the possibility of obfuscation poses a potential problem for detecting deceptive alignment. However, in the case of deceptive alignment, we also have access to the entire training process, including the training dataset, which may be enough information for us to detect the trigger despite any potential obfuscation. By analogy, in our construction in Theorem 5.5, it would be easy for the defender to detect the trigger if they had access to the unpunctured key K that was used in the construction of f^*.

This motivates the study of variants of our formal notion of defendability that constrain the attacker in different ways, or provide more assistance to the defender. To better capture the analogy with deceptive alignment, we would be excited to see research into variants that provide the defender with more information about how the function they are given was constructed, to see if this makes defense computationally feasible.

8 Conclusion

We have introduced a formal notion of defendability against backdoors in which the attacker's strategy must work for a randomly-chosen trigger. Despite its simplicity, this notion gives rise to a rich array of strategies. In the absence of computational constraints, defense is exactly as hard as learning. Meanwhile, in the presence of computational constraints, defense

is strictly easier than learning, but impossible for function classes that are rich enough to support obfuscation. We are excited to see future work that further explores the exact relationship between defense and learning.

References

1 Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *Annual international cryptology conference*, pages 1–18. Springer, 2001. doi:10.1007/3-540-44647-8_1.

2 Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989. doi:10.1145/76359.76371.

3 Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In *Advances in Cryptology-ASIACRYPT 2013: 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II 19*, pages 280–300. Springer, 2013. doi:10.1007/978-3-642-42045-0_15.

4 Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In *International workshop on public key cryptography*, pages 501–519. Springer, 2014. doi:10.1007/978-3-642-54631-0_29.

5 Sébastien Bubeck, Yin Tat Lee, Eric Price, and Ilya Razenshteyn. Adversarial examples from computational constraints. In *International Conference on Machine Learning*, pages 831–840. PMLR, 2019. URL: http://proceedings.mlr.press/v97/bubeck19a.html.

6 Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR, 2019. URL: http://proceedings.mlr.press/v97/cohen19c.html.

7 Jacob Dumford and Walter Scheirer. Backdooring convolutional neural networks via targeted weight perturbations. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–9. IEEE, 2020. doi:10.1109/IJCB48548.2020.9304875.

8 Sanjam Garg, Somesh Jha, Saeed Mahloujifar, and Mahmoody Mohammad. Adversarially robust learning could leverage computational hardness. In *Algorithmic Learning Theory*, pages 364–385. PMLR, 2020. URL: http://proceedings.mlr.press/v117/garg20a.html.

9 Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM (JACM)*, 33(4):792–807, 1986. doi:10.1145/6490.6503.

10 Oded Goldreich and Leonid A Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32, 1989. doi:10.1145/73007.73010.

11 Shafi Goldwasser, Michael P Kim, Vinod Vaikuntanathan, and Or Zamir. Planting undetectable backdoors in machine learning models. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 931–942. IEEE, 2022.

12 Shafi Goldwasser, Jonathan Shafer, Neekon Vafa, and Vinod Vaikuntanathan. Oblivious defense in ML models: Backdoor removal without detection. *arXiv preprint*, 2024. arXiv:2411.03279.

13 Steve Hanneke, Amin Karbasi, Mohammad Mahmoody, Idan Mehalel, and Shay Moran. On optimal learning under targeted data poisoning. *Advances in Neural Information Processing Systems*, 35:30770–30782, 2022.

14 David Haussler, Nick Littlestone, and Manfred K Warmuth. Predicting {0, 1}-functions on randomly drawn points. *Information and Computation*, 115(2):248–292, 1994.

15 Sanghyun Hong, Nicholas Carlini, and Alexey Kurakin. Handcrafted backdoors in deep neural networks. *Advances in Neural Information Processing Systems*, 35:8068–8080, 2022.

16 Evan Hubinger, Chris van Merwijk, Vladimir Mikulik, Joar Skalse, and Scott Garrabrant. Risks from learned optimization in advanced machine learning systems. *arXiv preprint*, 2019. arXiv:1906.01820.

17 Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 60–73, 2021. doi:10.1145/3406325.3451093.

18 Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. Intrinsic certified robustness of bagging against data poisoning attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35(9), pages 7961–7969, 2021. doi:10.1609/AAAI.V35I9.16971.

19 Adam Tauman Kalai and Shang-Hua Teng. Decision trees are PAC-learnable from most product distributions: a smoothed analysis. *arXiv preprint*, 2008. arXiv:0812.0933.

20 Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography: principles and protocols*. Chapman and hall/CRC, 2007.

21 Michael Kearns and Leslie Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the ACM (JACM)*, 41(1):67–95, 1994. doi:10.1145/174644.174647.

22 Michael J Kearns and Umesh Vazirani. *An introduction to computational learning theory*. MIT press, 1994.

23 Alaa Khaddaj, Guillaume Leclerc, Aleksandar Makelov, Kristian Georgiev, Hadi Salman, Andrew Ilyas, and Aleksander Madry. Rethinking backdoor attacks. In *International Conference on Machine Learning*, pages 16216–16236. PMLR, 2023. URL: https://proceedings.mlr.press/v202/khaddaj23a.html.

24 Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 669–684, 2013. doi:10.1145/2508859.2516668.

25 Alexander Levine and Soheil Feizi. Deep partition aggregation: Provable defense against general poisoning attacks. *arXiv preprint*, 2020. arXiv:2006.14768.

26 Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

27 Ryan O'Donnell. Analysis of Boolean functions. *arXiv preprint*, 2021. arXiv:2105.10386.

28 Chris Olah. Mechanistic interpretability, variables, and the importance of interpretable bases. *Transformer Circuits Thread*, 2022. URL: https://www.transformer-circuits.pub/2022/mech-interp-essay.

29 Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 475–484, 2014. doi:10.1145/2591796.2591825.

30 Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984. doi:10.1145/1968.1972.

A Alternative version of Theorem 4.1 using a Boltzmann posterior

In this section we prove a slightly weaker version of Theorem 4.1 using a detection strategy that does not involve learning. Instead of using distillation plus ensembling as in the proof of Theorem 4.1, we employ an alternative method of regularization that involves resampling the given function from a Boltzmann posterior. This gives rise to an extra logarithmic factor in the confidence of the detection algorithm, but otherwise gives the same bound, and in particular the threshold for detecting backdoors in a class of VC dimension d is still $\varepsilon = \Theta\left(\frac{1}{d}\right)$.

We are grateful to Dmitry Vaintrob for an earlier version of these results.

The remainder of this section has been omitted from this version due to the page limit. For the full version, see: https://arxiv.org/abs/2409.03077

B Proof of Theorem 4.1

▶ **Theorem 4.1.** *Let $\mathcal{F}$ be a representation class over $\{0,1\}^n$ and let $\varepsilon > 0$. Then the most (i.e., supremum) confidence with which $\mathcal{F}$ is ε-defendable is*

$$\max\left(\tfrac{1}{2}, 1 - \Theta\left(\mathrm{VC}\left(\mathcal{F}\right)\varepsilon\right)\right)$$

as $\mathrm{VC}\left(\mathcal{F}\right) \to \infty$.

Proof. Write $d = \mathrm{VC}\left(\mathcal{F}\right)$. We will show that, more specifically, for $d \geq 1$:

(a) For any detection strategy there is an adversary limiting their win probability to at most $\max\left(\frac{1}{2}, 1 - \frac{1}{2}\left(d-1\right)\varepsilon\right)$.

(b) There is a detection strategy that wins against any adversary with probability at least $\max\left(\frac{1}{2}, 1 - 15d\varepsilon\right)$.

For part a, let S be a set of size d shattered by $\mathcal{F}$, and let $s \in S$ be any point. The adversary proceeds by taking $\mathcal{D}$ to assign probability $\min\left(\frac{1}{d-1}, \varepsilon\right)$ to each point of $S \setminus \{s\}$ and the remaining $1-p$ probability to the point s, where $p := \min\left(1, \left(d-1\right)\varepsilon\right)$. The adversary then takes f to be one of the 2^{d-1} witnesses for the shattering that assigns 0 to the point s. We may treat f as if it were chosen uniformly at random, since the detection strategy's worst-case performance can be no better than its average-case performance. Finally, given x^*, the adversary takes f^* to be the witness that agrees with f exactly on the set $S \setminus \{x^*\}$ if $x^* \neq s$, and otherwise takes $f^* = f$. It is easy to check that f^* is ε-valid if and only if $x^* \neq s$. Furthermore, (f, x) and (f^*, x^*) are identically distributed for $x \sim \mathcal{D}$. Hence the best a detection strategy can do is to output Acc when given the point s, and otherwise the strategy can do no better than chance. So the detection strategy's overall win probability is at most $(1-p) + \frac{p}{2} = 1 - \frac{p}{2}$, as required.

For part b, note first that the detection strategy can achieve a win probability of at least $\frac{1}{2}$ simply by guessing randomly, so we may assume that $1 - 15d\varepsilon > \frac{1}{2}$. To construct the detection strategy, by Haussler–Littlestone–Warmuth, take a prediction strategy for $\mathcal{F}$ with sample size $m-1$ and error rate at most $\frac{d}{m}$, for some positive integer m to be chosen later. The detection strategy then works as follows:

- Given (f', x') as input (which could be either (f, x) or (f^*, x^*)), we use the prediction strategy to make a prediction z for $f'\left(x'\right)$, which we in turn think of as a prediction for $f\left(x'\right)$. The idea is that if we can successfully predict $f\left(x'\right)$, then we can distinguish the two cases by comparing this to $f'\left(x'\right)$, since $f\left(x\right) = f\left(x\right)$ but $f\left(x^*\right) \neq f^*\left(x^*\right)$ if f^* is ε-valid.
- We repeat this procedure r times with independent samples to obtain predictions $z^{(1)}, \ldots, z^{(r)}$ for some positive integer r to be chosen later.
- Finally, we take a majority vote: if more than half of $z^{(1)}, \ldots, z^{(r)}$ are different from $f'\left(x'\right)$, then we output Rej, and otherwise we output Acc.

To lower bound the detection strategy's win probability, consider first the case in which (f, x) is passed to the detection strategy. In this case the probability that $z \neq f\left(x\right)$ is at most the error rate $\frac{d}{m}$. Hence by Markov's inequality and linearity of expectation, the probability that more than half of $z^{(1)}, \ldots, z^{(r)}$ are different from $f\left(x\right)$ is at most $\frac{2d}{m}$, giving the detection strategy a failure probability of at most $\frac{2d}{m}$. If instead (f^*, x^*) is passed to the detection strategy, then there are two possible reasons why we could have $z \neq f\left(x^*\right)$: either f and f^* disagree on at least one of the $m-1$ samples provided by the example oracle, or the prediction strategy fails to predict $f\left(x^*\right)$ even if it is provided with the value of f on all

of these $m-1$ samples. Write $\mathbb{1}^{(i)}_{f \text{ and } f^* \text{ disagree}}$ and $\mathbb{1}^{(i)}_{\text{prediction fails}}$ for the indicator functions of these two events respectively during the ith trial for $i \in \{1, \ldots, r\}$. Then the probability that the detection strategy fails is

$$\mathbb{P}\left(\frac{1}{r}\sum_{i=1}^{r} \mathbb{1}_{z^{(i)} \neq f(x^*)} \geq \frac{1}{2}\right) \leq \mathbb{P}\left(\frac{1}{r}\sum_{i=1}^{r} \mathbb{1}^{(i)}_{f \text{ and } f^* \text{ disagree}} \geq \frac{1}{4}\right) + \mathbb{P}\left(\frac{1}{r}\sum_{i=1}^{r} \mathbb{1}^{(i)}_{\text{prediction fails}} \geq \frac{1}{4}\right).$$

The second term on the right-hand side is at most $\frac{4d}{m}$ by another application of Markov's inequality and linearity of expectation. To bound the first term on the right-hand side, we may assume without loss of generality that f^* is ε-valid, which implies that $p := \mathbb{P}\left(\mathbb{1}^{(i)}_{f \text{ and } f^* \text{ disagree}} = 1\right) \leq (m-1)\,\varepsilon$, by the union bound. Taking m to be the unique positive integer with $\frac{1}{5\varepsilon} < m \leq \frac{1}{5\varepsilon} + 1$, we have $p \leq \frac{1}{5}$. Since these r indicator functions are independent, the first term on the right-hand side is at most $\exp\left(-2\left(\frac{1}{4} - \frac{1}{5}\right)^2 r\right) = \exp\left(-\frac{r}{200}\right)$ by Hoeffding's inequality, which can be made arbitrarily small by taking r to be sufficiently large. Putting everything together, the detection strategy's best overall win probability is at least

$$\tfrac{1}{2}\left(\left(1 - \tfrac{2d}{m}\right) + \left(1 - \tfrac{4d}{m}\right)\right) = 1 - \tfrac{3d}{m} > 1 - 15\varepsilon d,$$

as required. ◀

C Proof of Theorem 5.3

▶ **Theorem 5.3.** *In the random oracle model of computation, with probability $1 - O(2^{-n^c})$ for all c over the choice of random oracle, there is a polynomially evaluatable representation class over $\mathcal{X} = \{0,1\}^n$ that is efficiently defendable but not efficiently PAC learnable.*

Proof. Write $\textsc{Rand} : \{0,1\}^* \to \{0,1\}$ for the random oracle. Let $\mathcal{K} = \{0,1\}^n$ be a set of "keys", and let $\mathcal{F} = \{x \mapsto \textsc{Rand}(K \parallel x) \mid K \in \mathcal{K}\}$, where $\parallel$ denotes string concatenation. We will show that, over the choice of random oracle, $\mathcal{F}$ is efficiently defendable with probability $1 - o(2^{-n^c})$ as $n \to \infty$ for all $c \geq 1$, but not efficiently PAC learnable with probability $1 - o\left(\exp\left(-2^{n-4}\right)\right)$ as $n \to \infty$. The result then follows by the union bound.

For defendability, let $\delta > 0$ and $c \geq 1$. We will show that, with probability $1 - o(2^{-n^c})$ over the choice of random oracle, if $\varepsilon = o\left(\frac{\delta}{n^c}\right)$ then $\mathcal{F}$ is ε-defendable with confidence $1 - \delta$ using the trivial detection strategy that always outputs $\textsc{Acc}$. To see this, let $\mathcal{D}$ be the distribution over $\mathcal{X}$ chosen by the adversary, and for $x \in \mathcal{X}$ write $\mathbb{P}_{\mathcal{D}}(x)$ for $\mathbb{P}_{x' \sim \mathcal{D}}(x' = x)$. In order for the adversary to be able to choose a backdoored function that is ε-valid with high enough probability over the choice of backdoor trigger, the adversary must choose $\mathcal{D}$ so that

$$S_1 := \sum_{\substack{x \in \mathcal{X} \\ \mathbb{P}_{\mathcal{D}}(x) \leq \varepsilon}} \mathbb{P}_{\mathcal{D}}(x) > 2\delta.$$

Note also that

$$S_2 := \sum_{\substack{x \in \mathcal{X} \\ \mathbb{P}_{\mathcal{D}}(x) \leq \varepsilon}} \mathbb{P}_{\mathcal{D}}(x)^2 \leq \varepsilon S_1.$$

Now for any $K, K^* \in \mathcal{K}$ and any $\varepsilon < \frac{1}{2}S_1$,

$$\begin{aligned}
&\mathbb{P}_{\text{RAND}}\left(\mathbb{P}_{x\sim D}\left(\text{RAND}\left(K \parallel x\right) \neq \text{RAND}\left(K^* \parallel x\right)\right) \leq \varepsilon\right) \\
&\leq \mathbb{P}_{\text{RAND}}\left(\sum_{\substack{x\in\mathcal{X} \\ \mathbb{P}_{\mathcal{D}}(x)\leq\varepsilon}} \mathbb{P}_{\mathcal{D}}\left(x\right) \mathbb{1}_{\text{RAND}(K\parallel x)\neq\text{RAND}(K^*\parallel x)} \leq \varepsilon\right) \\
&\leq \exp\left(-\frac{2\left(\frac{1}{2}S_1 - \varepsilon\right)^2}{S_2}\right) && \text{(by Hoeffding's inequality)} \\
&\leq \exp\left(-\frac{S_1}{2\varepsilon} - \frac{2\varepsilon}{S_1} + 2\right) && \text{(since } S_2 \leq \varepsilon S_1\text{)} \\
&< \exp\left(-\frac{\delta}{\varepsilon} - 2\varepsilon + 2\right) && \text{(since } 2\delta < S_1 \leq 1\text{)}.
\end{aligned}$$

Hence by the union bound, it is impossible for the adversary to choose any $K, K^* \in \mathcal{K}$ with $\mathbb{P}_{x\sim D}\left(\text{RAND}\left(K \parallel x\right) \neq \text{RAND}\left(K^* \parallel x\right)\right) \leq \varepsilon$ with probability at least

$$1 - \binom{|\mathcal{K}|}{2}\exp\left(-\frac{\delta}{\varepsilon} - 2\varepsilon + 2\right) > 1 - \exp\left((2n-1)\log\left(2\right) - \frac{\delta}{\varepsilon} - 2\varepsilon + 2\right).$$

When this event happens the detection strategy always wins, and if $\varepsilon = o\left(\frac{\delta}{n^c}\right)$ then this probability is $1 - o(2^{-n^c})$, as required.

For PAC learnability, we will show that, with probability $1 - o\left(\exp\left(-2^{n-4}\right)\right)$ over the choice of random oracle, there is no PAC learning algorithm for $\mathcal{F}$, even with access to the random oracle, that runs in time $f\left(n\right) = 1.5^n$ with confidence parameter $\frac{1}{4}$ and error parameter $\frac{1}{4}$, say. To see this, note that, for all sufficiently large n, there are fewer than $2^{f(n)}$ possible hypotheses that such an algorithm could output, assuming by convention that algorithms can process at most one bit per unit time. Now for each of these hypotheses h, if $\mathcal{D}$ is the uniform distribution over $\mathcal{X}$ and $K \in \mathcal{K}$, then

$$\mathbb{P}_{\text{RAND}}\left(\mathbb{P}_{x\sim D}\left(\text{RAND}\left(K \parallel x\right) \neq h\left(x\right)\right) \leq \tfrac{1}{4}\right) \leq \exp\left(-2\left(\tfrac{1}{2} - \tfrac{1}{4}\right)^2 |\mathcal{X}|\right) = \exp\left(-2^{n-3}\right)$$

by Hoeffding's inequality. Hence by the union bound, the probability that none of these hypotheses has generalization error at most $\frac{1}{4}$ is at least

$$1 - 2^{f(n)}\exp\left(-2^{n-3}\right) = 1 - \exp\left(1.5^n\log\left(2\right) - 2^{n-3}\right) = 1 - o\left(\exp\left(-2^{n-4}\right)\right).$$

When this event happens the PAC learning algorithm fails with probability $1 > \frac{1}{4}$, as required. ◀

D Proof of Theorem 6.1

▶ **Theorem 6.1.** *The representation class $\mathcal{F}$ of decision trees over $\mathcal{X} = \{0,1\}^n$ of size at most s, where s is polynomial in n, is efficiently uniform-defendable using a detection strategy that makes 0 calls to the example oracle and runs in time $O\left(\text{time taken to evaluate a single decision tree}\right)$.*

Proof. Given $f \in \mathcal{F}$ and $x \in \mathcal{X}$, write $\text{DEPTH}\left(f, x\right)$ for the number of distinct input variables that appear along the path from root to leaf when f is evaluated at x.

Let $\delta > 0$, and take $\varepsilon < \frac{\delta^2}{s^2}$. We claim that, as long as the adversary takes $\mathcal{D}$ to be uniform distribution, $\mathcal{F}$ is ε-defendable with confidence $1-\delta$ using the detection strategy that, given (f', x') as input, outputs Rej if $2^{-\textsc{Depth}\left(f', x'\right)} \leq \frac{\delta}{s}$ and Acc otherwise. This suffices, since evaluating $\textsc{Depth}\left(f', x'\right)$ takes the same time as evaluating $f'\left(x'\right)$, up to a constant factor.

To see this, given $f \in \mathcal{F}$ and $x \in \mathcal{X}$, write

$$\textsc{Leaf}\left(f, x\right) = \left\{x' \in \mathcal{X} \mid x' \text{ and } x \text{ reach the same leaf of } f\right\},$$

and observe that

$$\mathbb{P}_{\mathcal{D}}\left(\textsc{Leaf}\left(f, x\right)\right) = 2^{-\textsc{Depth}(f,x)},$$

where we have written $\mathbb{P}_{\mathcal{D}}\left(S\right)$ as shorthand for $\mathbb{P}_{x\sim\mathcal{D}}\left(x \in S\right)$. The key claim is that, for any $f \in \mathcal{F}$ and any $x^* \in \mathcal{X}$, if $f^* \in \mathcal{F}$ is ε-valid, then

$$2^{-\textsc{Depth}(f,x^*)-\textsc{Depth}(f^*,x^*)} \leq \mathbb{P}_{\mathcal{D}}\left(\textsc{Leaf}\left(f, x^*\right) \cap \textsc{Leaf}\left(f^*, x^*\right)\right) \leq \varepsilon.$$

The first inequality is a consequence of the fact that the number of distinct variables that appear when either f or f^* is evaluated at x^* is at most $\textsc{Depth}\left(f, x^*\right) + \textsc{Depth}\left(f^*, x^*\right)$, and makes essential use of the fact that $\mathcal{D}$ is uniform (or at the very least, a product distribution). The second inequality follows from the definition of ε-valid.

Now choose f, x^*, f^* and x as in the backdoor detection game. Suppose first that (f, x) is passed to the detection strategy. Then with probability at least $1-\delta$, we have $2^{-\textsc{Depth}(f,x)} = \mathbb{P}_{\mathcal{D}}\left(\textsc{Leaf}\left(f, x\right)\right) > \frac{\delta}{s}$, since f has at most s leaves. Hence the detection strategy outputs Acc with probability at least $1-\delta$. Now suppose instead that (f^*, x^*) is passed to the detection strategy. Then with probability at least $1-\delta$, we have $2^{-\textsc{Depth}(f,x^*)} > \frac{\delta}{s}$, by a similar argument. But if f^* is ε-valid, then this implies that $2^{-\textsc{Depth}(f^*,x^*)} < \varepsilon/\frac{\delta}{s} < \frac{\delta}{s}$, by the key claim. Hence the detection strategy outputs Rej with probability at least $1-\delta$. So the detection strategy's overall win probability is at least $1-\delta$, as required. ◀

Data Reconstruction: When You See It and When You Don't

Edith Cohen
Google Research, Mountain View, CA, USA
Tel Aviv University, Israel

Haim Kaplan
Tel Aviv University, Israel
Google Research, Tel Aviv, Israel

Yishay Mansour
Tel Aviv University, Israel
Google Research, Tel Aviv, Israel

Shay Moran
Technion, Haifa, Israel
Google Research, Tel Aviv, Israel

Kobbi Nissim
Georgetown University, Washington, DC, USA
Work done while at Google Research,
Tel Aviv, Israel

Uri Stemmer
Tel Aviv University, Israel
Google Research, Tel Aviv, Israel

Eliad Tsfadia
Georgetown University, Washington, DC, USA

Abstract

We revisit the fundamental question of formally defining what constitutes a *reconstruction attack*. While often clear from the context, our exploration reveals that a precise definition is much more nuanced than it appears, to the extent that a single all-encompassing definition may not exist. Thus, we employ a different strategy and aim to "sandwich" the concept of reconstruction attacks by addressing two complementing questions: (i) What conditions guarantee that a given system is protected against such attacks? (ii) Under what circumstances does a given attack clearly indicate that a system is not protected? More specifically,

- We introduce a new definitional paradigm – *Narcissus Resiliency* – to formulate a security definition for protection against reconstruction attacks. This paradigm has a self-referential nature that enables it to circumvent shortcomings of previously studied notions of security. Furthermore, as a side-effect, we demonstrate that Narcissus resiliency captures as special cases multiple well-studied concepts including differential privacy and other security notions of one-way functions and encryption schemes.
- We formulate a link between reconstruction attacks and *Kolmogorov complexity*. This allows us to put forward a criterion for evaluating when such attacks are convincingly successful.

2012 ACM Subject Classification Security and privacy → Human and societal aspects of security and privacy

Keywords and phrases differential privacy, reconstruction

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.39

Related Version *Full Version*: `https://arxiv.org/abs/2405.15753` [17]

Funding *Edith Cohen*: Partially supported by Israel Science Foundation (grant 1595/19 and 1156/23).
Haim Kaplan: Partially supported by Israel Science Foundation (grant 1595/19 and 1156/23) and the Blavatnik Family Foundation.
Yishay Mansour: Partially funded from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 882396), by the Israel Science Foundation (grant number 993/17), Tel Aviv University Center for AI and Data Science (TAD), and the Yandex Initiative for Machine Learning at Tel Aviv University.
Shay Moran: A Robert J. Shillman Fellow; he acknowledges support by ISF grant 1225/20, by BSF grant 2018385, by an Azrieli Faculty Fellowship, by Israel PBC-VATAT, by the Technion Center for Machine Learning and Intelligent Systems (MLIS), and by the European Union (ERC, GENERALIZATION, 101039692). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.



© Edith Cohen, Haim Kaplan, Yishay Mansour, Shay Moran, Kobbi Nissim, Uri Stemmer, and Eliad Tsfadia;
licensed under Creative Commons License CC-BY 4.0

Kobbi Nissim: Partially funded by NSF grant No. 2217678 and by a gift to Georgetown University.
Uri Stemmer: Partially supported by the Israel Science Foundation (grant 1419/24) and the Blavatnik Family foundation.
Eliad Tsfadia: Supported by a gift to Georgetown University.

Acknowledgements The authors would like to thank Noam Mazor for useful discussions about Kolmogorov complexity.

1 Introduction

Reasoning about data privacy is crucial in today's data-driven world. This includes the design of privacy-enhancing technologies aimed at protecting privacy, as well as identifying vulnerabilities of existing methods by conducting "privacy attacks". The most severe family of attacks is, arguably, *reconstruction attacks*, where an adversary takes what appears to be benign statistics and reconstructs significant portions of the sensitive data [19]. Such attacks have helped shape the theory of data privacy as well as expose vulnerabilities in existing real-world systems.

However, prior works did not coalesce around a single *definition of reconstruction* and instead considered several context-dependent definitions. Although these definitions made sense in the context in which they were introduced, they do not necessarily carry over to other settings, as we will later explain. Motivated by this, our work is set to answer the following meta-question:

▶ **Question 1.** *What is reconstruction?*

This question is more nuanced than it initially appears, as will become clear later. It seems that for every attempt to define reconstruction mathematically, there are cases that either fit the definition but do not "feel like" reconstruction, or vice versa. As a result, we do not know whether a precise answer to Question 1 exists, and leave it as an open question for future work. In this work, we aim to approach Question 1 by "sandwiching" the concept of reconstruction and studying the following two questions:

▶ **Question 2.** *What conditions are sufficient to ensure that a given system is protected against reconstruction attacks?*

▶ **Question 3.** *Under what circumstances does an attack clearly indicate that a given system is vulnerable?*

This is reminiscent of the current state of affairs in the related concept of *differential privacy (DP)* [20]: If an algorithm satisfies DP (with small enough privacy parameters), then it is guaranteed to be "safe" in terms of its privacy implications. However, the fact that an algorithm does not satisfy DP does not immediately mean that it is unsafe privacy-wise. Rather, to convince that an algorithm is unsafe, one usually needs to demonstrate an actual attack, such as a reconstruction or a membership attack. In other words, in the context of data privacy, we currently do not have a good definition of what constitutes "privacy"; only definitions of what it means to "protect" privacy, and definitions for what it means to "attack" an algorithm in order to show a privacy breach. We embrace this way of thinking and leverage it towards formally reasoning about reconstruction, as stated in Questions 2 and 3. We elaborate on each of these two questions separately, in Sections 1.1 and 1.2, respectively.

1.1 Protecting against reconstruction

Consider an algorithm $\mathcal{M}$ that takes a dataset S and returns an output y. For example, the dataset S might contain images, and the output y may be an image generative model. This output y is then given to a "privacy attacker" $\mathcal{A}$ whose goal is to output a "reconstruction" z of elements from S. Informally, we want to say that algorithm $\mathcal{M}$ *prevents reconstruction attacks* if y is "hard to invert" in the sense that y does not help the attacker in "recovering" elements from S. However, there are two immediate issues that require attention here:

1. What it means to "recover elements" is context-dependent. For example, our criteria for when one image "recovers" another image would likely differ from our criteria for when a text file "recovers" another text file. As we aim for a general treatment, we abstract this criteria away in the form of a *relation* R, where $R(S, z) = 1$ if and only if z is a "valid reconstruction" of S (or of some data point in S). In our example with the images, this could mean that z is an image which is "close enough" to one of the images in S. Alternatively, depending on the context, this could also mean that z is a collection of 100 images of which one is "close enough" to an image in S, or it could mean that z is a vector of 100 images, each of them is "somewhat close" to a distinct image in S, etc.
2. The underlying distribution of the data matters a lot. For example, consider a scenario where every sufficiently large image dataset contains the Mona Lisa. An "attack" that outputs an image that is similar or identical to the Mona Lisa is not necessarily a successful attack, as this could be accomplished without even accessing the generative model y.

These considerations led [4] and [18] to present definitions with the following flavor:

▶ **Definition 4** ([4, 18]). *Let $\mathcal{X}$ be a data domain, let $\mathcal{D}$ be a distribution over datasets containing elements from $\mathcal{X}$, and let $R : \mathcal{X}^* \times \{0,1\}^* \to \{0,1\}$ be a reconstruction criterion. Algorithm $\mathcal{M}$ is $(\varepsilon, \delta, \mathcal{D})$-$R$-reconstruction-robust if for all attackers $\mathcal{A}$ it holds that*

$$\Pr_{\substack{S\leftarrow\mathcal{D}\\ y\leftarrow\mathcal{M}(S)\\ z\leftarrow\mathcal{A}(y)}}[R(S,z)=1] \leq e^{\varepsilon}\cdot\sup_{z^*}\Pr_{T\leftarrow\mathcal{D}}[R(T,z^*)=1]+\delta. \tag{1}$$

In words, in Definition 4 the attacker's probability of success is compared to a *baseline* which is the probability of success of the best *trivial attacker* that simply chooses a fixed element z^* (independently of the dataset). To contradict the security of algorithm $\mathcal{M}$, the attacker must succeed with a probability noticeably higher than the baseline (depending on the parameters ε and δ). Note that the aforementioned attacker that "recovered" the Mona Lisa would *not* contradict the security of $\mathcal{M}$. The reason is that if every (large enough) dataset sampled from $\mathcal{D}$ would contain the Mona Lisa, then in Definition 4 we could take z^* to be the Mona Lisa, and hence the right hand size of Inequality (1) would be one.[1]

This introduces a serious problem: once the baseline probability is 1 then *no adversary* could ever contradict Inequality (1), even adversaries that do achieve meaningful reconstructions. To illustrate the issue, suppose that the dataset contains both "canonical" photos (such as the Mona Lisa or canonical photos of US presidents), as well as "sensitive" photos of ordinary individuals. While recovering a canonical photo should not be considered a successful attack, recovering a "sensitive" photo definitely should. Formally, suppose that every image in S is sampled independently as follows: with probability 1/2 return the Mona Lisa, and otherwise return a photo of a random citizen. If under these conditions $\mathcal{A}(\mathcal{M}(S))$

[1] For this example we assume that if $z^* \in T$ then $R(T, z^*) = 1$.

is able to recover a photo of an ordinary individual from S, then that should be considered a reconstruction attack. But this is not captured by Definition 4 as the performance of $\mathcal{A}$ is compared with a baseline where the Mona Lisa is recovered. More generally, a sever flaw of Definition 4 is that once the baseline probability is ≈ 1 then no adversary could ever contradict Inequality (1).

1.1.1 Towards a new definition

A takeaway from the above discussion is that comparing all adversaries to the same fixed baseline can create a problem. In particular, adversaries that identify "special" elements in S (which are unlikely to appear in fresh datasets) should be distinguished from adversaries that identify "trivial" elements in S. We now make an attempt to incorporate this into the definition. As we will see, this attempt has different shortcomings.

▶ **Definition 5.** *Let $\mathcal{X}$ be a data domain, let $\mathcal{D}$ be a distribution over datasets containing elements from $\mathcal{X}$, and let $R : \mathcal{X}^* \times \{0,1\}^* \to \{0,1\}$. Algorithm $\mathcal{M}$ is $(\varepsilon, \tau, \mathcal{D})$-R-reconstruction-robust if for all attackers $\mathcal{A}$ it holds that*

$$\Pr_{\substack{S\leftarrow\mathcal{D}\\ y\leftarrow\mathcal{M}(S)\\ z\leftarrow\mathcal{A}(y)}}\Big[R(S,z)=1 \;\; \textit{and} \;\; \Pr_{T\leftarrow\mathcal{D}}[R(T,z)=1]\leq\tau\Big]\leq\varepsilon. \tag{2}$$

In words, with this definition the attacker's goal is to identify an element z such that (1) z is a valid reconstruction w.r.t. the dataset S; and (2) the same z is unlikely to be a valid reconstruction w.r.t. a fresh dataset T. Informally, $\Pr_{T\leftarrow\mathcal{D}}[R(T,z)=1]$ serves as a "conditional baseline" that adapts itself to the element z chosen by the attacker. Note that in the context of our example with the Mona Lisa, an attacker that "recovers" the Mona Lisa would not contradict Inequality (2), because $\Pr_{T\leftarrow\mathcal{D}}[R(T,z)=1]$ would be large and hence greater than the threshold τ. On the other hand, an attacker that manages to recover the photo of an ordinary individual from S (with large enough probability) would contradict Inequality (2). So Definition 5 achieves the desired behavior for this example.

A shortcoming of Definition 5 is that the values of ε and τ are necessarily context dependent. To illustrate this, we will now describe two situations where in one of them we need to set $\varepsilon, \tau < \frac{1}{2}$ in order for the definition to make sense, while in the other situation ε and τ must be larger than $\frac{1}{2}$.

- For the first situation, consider the following distribution $\mathcal{D}$ over datasets: With probability $\frac{1}{2}$ return a dataset containing random photos of ordinary citizens. Otherwise, return a dataset sampled similarly, except that one of its photos is replaced with the Mona Lisa. Under these conditions, we must set $\tau < \frac{1}{2}$ in order to circumvent the "trivial attack" using the Mona Lisa. Setting ε close to 0 seems reasonable under these conditions in order to guarantee that a meaningful reconstruction can happen only with small probability.
- For the second situation, consider a case where the dataset is a vector of n random bits, and the attacker's goal is to pinpoint a single entry from this vector and to guess it with high probability. Formally, in this example we interpret the outcome of the adversary as $z = (i, b) \in [n] \times \{0,1\}$, where $R(S,(i,b)) = 1$ if and only if $S[i] = b$. In this scenario, *every* adversary always satisfies $\Pr_{T\leftarrow\mathcal{D}}[R(T,z)=1] = 1/2$ and hence if we set $\tau < 1/2$ then every algorithm is safe w.r.t. to that definition, which is absurd. Therefore we may assume that $\tau \geq 1/2$. In this case the condition $\Pr_{T\leftarrow\mathcal{D}}[R(T,z)=1] \leq \tau$ holds for every adversary and hence is redundant and we are only left with the first condition. How about ε? Note that a trivial adversary, which always guesses that the first bit is 1,

succeeds with probability $1/2$. Hence, ε must be $\geq 1/2$ or else no algorithm can be safe. To summarize, if $\tau < 1/2$ then every algorithm is safe, and else, if $\tau \geq 1/2$ and $\varepsilon < 1/2$ then no algorithm is safe. Hence, this condition is non-trivial only in the range $\varepsilon, \tau \geq 1/2$.

Intuitively, the issue about ε, τ being context dependent stems from their somewhat nonstandard semantics (especially the parameter τ). Ideally, we would want ε and τ to quantify some notion of "distance from optimality" (similarly to common definitions in the literature, such as differential privacy). But this is not the case with Definition 5. Rather, the threshold τ dictates what we consider to be a "non-trivial" baseline. But "non-trivial" is context dependent, and hence so is τ.

1.1.2 A new Definitional paradigm: Narcissus resiliency – an adversary trying to beat itself in its own game

So far, we have established that: (1) We need to compare the adversary's success probability to a baseline; (2) This baseline cannot be fixed for all adversaries; and (3) We want to circumvent the need for a hyperparameter controlling what "non-trivial" is, as this is likely to be context dependent. To tackle this, we present a new definitional paradigm which we call *Narcissus resiliency*. In this paradigm, instead of explicitly setting and tuning the baseline, we *let the attacker be its own baseline*.[2] More specifically, the attacker's success probability on the real dataset is compared with its success probability in a baseline setting where it does not get any information about the real dataset. Furthermore, the attacker must use the same strategy in both settings. This is enforced by preventing the attacker from receiving any information that can help it distinguish whether it is examined with respect to real dataset S or a fresh dataset T sampled from the same distribution. Hence, a Narcissus attacker needs to choose a compromise strategy: on one hand it needs to be strong (so it succeeds with high probability when it is applied to S) and at the same time it needs to be weak (so it succeeds with low probability when it is applied to T). The formal definition follows:[3]

▶ **Definition 6** (Narcissus resiliency). *Let $\mathcal{X}$ be a data domain, let $\mathcal{F}$ be a family of distributions over datasets containing elements from $\mathcal{X}$, and let $R : \mathcal{X}^* \times \{0,1\}^* \to \{0,1\}$. Algorithm $\mathcal{M}$ is $(\varepsilon, \delta, \mathcal{F})$-R-Narcissus-resilient if for all $\mathcal{D} \in \mathcal{F}$ and for all attackers $\mathcal{A}$ it holds that*

$$\Pr_{\substack{S\leftarrow\mathcal{D}\\ y\leftarrow\mathcal{M}(S)\\ z\leftarrow\mathcal{A}(y)}}[R(S,z)=1] \leq e^{\varepsilon} \cdot \Pr_{\substack{S\leftarrow\mathcal{D}\\ T\leftarrow\mathcal{D}\\ y\leftarrow\mathcal{M}(S)\\ z\leftarrow\mathcal{A}(y)}}[R(T,z)=1] + \delta. \tag{3}$$

Note that in both experiments the adversary $\mathcal{A}$ is executed on a dataset S sampled exactly in the same way. Thus, the adversary behaves exactly the same way in both executions. Nevertheless, the adversary's goal is to "separate" the two experiments, where in the left experiment it aims to maximize the probability of reconstruction while in the right experiment it aims to minimize this probability.

▶ **Example 7.** Note that Definition 6 achieves the desired behavior in our example with the Mona Lisa: In order to contradict Inequality (3), the attacker must recover (with noticeable probability) the photo of a citizen from S.

[2] The name "Narcissus" is inspired by the Greek myth of Narcissus, who became obsessed with his own reflection. This alludes to how, in our proposed paradigm, the adversary competes against itself rather than external benchmarks.

[3] In the following definition we switch from a single data distribution $\mathcal{D}$ to a family of data distributions $\mathcal{F}$, where the mechanism is required to be "secure" w.r.t. every distribution in the family. This is standard, and can also be applied in the context of Definition 4.

▶ **Example 8.** Let the underlying dataset distribution $\mathcal{D}$ be uniform over n-bit vectors. Let $k \in [n]$ be a parameter and suppose that, for the sake of this example, the relation R is defined as follows. Given a dataset $S \in \{0,1\}^n$ and the outcome of the attacker z, parse z as a vector of k distinct indices $\tilde{I} = (i_1, \ldots, i_k) \in [n]^k$ and k values $\tilde{Z} = (\tilde{z}_1, \ldots, \tilde{z}_k) \in \{0,1\}^k$ and let $R(S, z) = 1$ if and only if $S|_{\tilde{I}} = \tilde{Z}$. That is, in this example, the adversary's goal is to pinpoint k coordinates from S and to guess them correctly. In order to contradict Inequality (3), the attacker must recover the values of k coordinates from S with probability noticeably higher than 2^{-k}.

We arrived at Definition 6 through the lens of data reconstruction. To provide further evidence supporting the paradigm of Narcissus resiliency, we show in Section 2 that Narcissus resiliency captures concepts from cryptography and privacy as special cases. This includes differential privacy, resiliency to membership inference attacks, security of one-way functions, security of encryption schemes, and more. We show that all of these concepts can be stated in the terminology of Narcissus resiliency, where an adversary is "trying to beat itself in its own game". Furthermore, the paradigm enables us to express strictly weaker protections compared to differential privacy that are still meaningful. To support this claim, in the full version [17] we provide an example of a very natural *deterministic* mechanism (which is clearly not differentially private) that estimates a counting query under Narcissus-resiliency w.r.t. the family of all i.i.d. distributions and a natural reconstruction predicate R.

1.2 Identifying reconstruction

So far we have discussed which types of mechanisms prevent reconstruction of the input data. Such definitions serve as important guidance for responsible algorithm design. However, to show that a mechanism is vulnerable to reconstruction in the real world, an attacker usually only has access to the output of the mechanism, which in the learning context is a model that can be seen as an interactive program, or even just as a fixed string (e.g., a real-world attacker would like to attack a given chat-bot model, and not the algorithm that created the model based on training data). Moreover, such attacker might not even be aware of the training procedure exactly.

Mathematically, in Section 1.1 we considered a setting where initially a dataset S is sampled, and then an outcome y is computed by an algorithm $\mathcal{M}$ based on S. Our definition of Narcissus resiliency 6 pinpoints our ultimate desire from the algorithm $\mathcal{M}$. Often, however, $\mathcal{M}$ is not explicit and we only have access to a fixed model y and a fixed training set S. So there is no clear distribution over S and it is not clear what is the process which was used to compute y from S. Still, in order to be able to assess whether a particular attack is successful, we would like to have a definition of what it means to "reconstruct" a particular S from a particular y. To emphasize this difference, when S and y are fixed and we want to classify a successful attack (rather than resiliency to such) we use the term "extract" instead of "reconstruct".

▶ **Question 9.** *How do we define that a* fixed *dataset S (or a* fixed *point in it $x \in S$) is extractable from a* fixed *string y?*

Intuitively, such "extraction" means that x is encoded in y. But capturing this intuition more formally turns out to be challenging. In this work, we offer the first formal definition that, we hope, will help understand what real-world attacks do, and what properties future attacks should highlight in order to argue about their quality.

Prior works presented several definitions with the following flavor:

▶ **Definition 10** ([14, 11]). *Let R be a relation such that $R(x,z)=1$ means that z is a valid extraction of x. A string x is* extractable *from a model y, if there exists an efficient program $\mathcal{A}$ such that $R(x,\mathcal{A}(y))=1$.*

Intuitively, the program $\mathcal{A}$ in Definition 10 serves as evidence that x is extractable from y, because given y it outputs z with $R(x,z)=1$. For example, if x appears at the i^{th} location of y, then the program that given y as input, outputs $y_i, y_{i+1}, \ldots, y_{i+|x|-1}$ satisfy the condition of Definition 10.

However, at the formal level, this definition is meaningless: Because $\mathcal{A}$ is chosen *after* the example x, then formally, every example x is "extractable" as there exists $\mathcal{A}$ that (ignores its input) and outputs x. [14] mentioned that in order to prevent such pathological cases, the program $\mathcal{A}$ should be *shorter* than x (and therefore $\mathcal{A}$ cannot "memorize" x). But this by itself is not enough. For example, consider the string $x = \underbrace{34\,34\,\ldots\,34}_{1000\text{ times}}$ and the program $\mathcal{A}$ "Print '34' 1000 times". Namely, this example illustrates that $\mathcal{A}$ being short is clearly not evidence for not being able to "memorize" x. Therefore, the length of $\mathcal{A}$ cannot be the only criteria for determining its validity, and there should be some connection between the length of the attack $\mathcal{A}$, and the "complexity" of the target output. Intuitively, if there is no way to encode the target output x using a short program (i.e., to compress it), then by demonstrating a short attacking program that is able to reveal x given y, proves that x is really extracted from the model y.

▶ **Question 11.** *How do we quantify the "complexity" of an outcome x?*

This question leads us to a classical complexity measure, called *Kolmogorov-Complexity*.

1.2.1 Defining Extraction via Kolmogorov Complexity

What makes the string 343434343434343 less random than 285628563123452? The notion of *Kolmogorov complexity* (in short, K-complexity), introduced by [56, 37, 15] in the 60s, provides an elegant method for measuring the amount of "randomness" in an individual string. Informally, the K-complexity of a string x, denoted by $K(x)$, is the length of the shortest program that outputs the string x. Intuitively, a string x has high K-complexity, if there is no short program that outputs it and halts.

One issue with K-complexity is that it is not well defined without specifying which programming language $\mathcal{L}$ do we use. In theoretical results (when constants do not matter), this complexity is usually defined w.r.t. a fixed *Universal Turing Machine*. But when constants matter, a more formal way to define it is as follows.

▶ **Definition 12** ($K_{\mathcal{L}}$-Complexity). *Let $\mathcal{L}$ be a programming language (e.g., Python). The $K_{\mathcal{L}}$-complexity of a string x, denoted by $K_{\mathcal{L}}(x)$, is the length of the shortest $\mathcal{L}$-program that outputs x and halts. Similarly, given a set of strings X, we denote by $K_{\mathcal{L}}(X)$ the length of the shortest $\mathcal{L}$-program that outputs an element in X and halts.*

Now that we have Definition 12 in hand, we are finally ready to define extraction.

▶ **Definition 13** (Our extraction definition, informal). *Let R be an extraction relation and $\mathcal{L}$ a programming language. We say that a string x is $(R,\mathcal{L})$-extractable from a string y iff there exists an $\mathcal{L}$-program $\mathcal{A}$ such that the following holds:*

1. *$\mathcal{A}(y)$ outputs z such that $R(x,z)=1$, and*
2. *$K_{\mathcal{L}}(\{z\colon R(x,z)=1\}) \gg |\mathcal{A}|$.*

We measure the quality *of the extraction by $1-\frac{|\mathcal{A}|}{K_{\mathcal{L}}(\{z\colon R(x,z)=1\})}$ (closer to 1 means a more significant extraction).*

Namely, x is "$(R, \mathcal{L})$-extractable" from y if there exists a *short* $\mathcal{L}$-program $\mathcal{A}$ that outputs an extraction (according to R) of x from y, while there is no short $\mathcal{L}$-program that extracts x without y. We note that extraction must be in a context of a specific programming language, since otherwise, for every x and y we could always find a "special" programming language $\mathcal{L}_{x,y}$ such that x is $\mathcal{L}_{x,y}$-extractable from y.[4] Satisfying this definition w.r.t. a *standard* programming language (e.g., Python, Java, etc) should indeed be considered as a valid extraction.

Definition 13 can easily be extended to datasets S (rather than a single example x), and in Section A we also consider a probabilistic version of the $K_{\mathcal{L}}$-complexity that allows some error probability, and redefine Definition 13 w.r.t. this version.

To provide further evidence supporting Definition 13, we demonstrate in Section A.3 how three different types of real-world attacks [14, 30, 11] can be explained using the terminology of Definition 13.

1.2.2 Narcissus-Resiliency prevents non-trivial extraction of training data

In Section 3 we show that if $\mathcal{M}$ is $(\varepsilon, \delta, \mathcal{F})$-$R$-Narcissus-resilient then it prevents extraction in the following sense. Fix a distribution $\mathcal{D} \in \mathcal{F}$, sample two independent datasets S and T from $\mathcal{D}$, and compute $y \leftarrow \mathcal{M}(S)$. We treat S as the "real" dataset and T as a "shadow" dataset. Suppose that there is no adversary $\mathcal{B}$ that given y can find a short program $\mathcal{A}$ such that $R(T, \mathcal{A}(y)) = 1$. Informally, this means that extracting information about the shadow dataset T without receiving any information about it is hard. Then, if $\mathcal{M}$ is Narcissus-resilient, there is also no adversary that achieves this w.r.t. S (even though the adversary gets y which was computed based on S).

1.2.3 Verifying the validity of reconstruction attacks

The main limitation of Definition 13 is that Item 2 (lower-bounding the Kolmogorov Complexity) is not verifiable, as computing the K-complexity is an intractable problem. We could consider a tractable version of it, called *time-bounded* K-complexity, where given a parameter t we consider only programs that halt within t steps ([37, 55, 57, 36]). This, however, would only relax intractability to inefficiency, which does not help in our context where we would like to verify a reconstruction attack efficiently. In practice, we can only use Heuristics to gain some level of confidence. E.g., to apply many well-known compression algorithms on x and check that all of them results with a compressed representation that is much longer than $|\mathcal{A}|$. Yet, we remark that any such Heuristic can fail to determine some highly compressible patterns, and as we demonstrate in Section A.4.1, this problem is indeed inherent assuming that a basic cryptography primitive (pseudo-random generator) exists.

1.3 Additional related works

Memorization

Perhaps the most basic privacy violation is *memorization*. [25, 24, 6, 7, 43, 1] theoretically study the necessity of memorization in learning. Roughly speaking, they show that for some tasks, the output of any accurate algorithm must have large *mutual information* with the training data. A similar result of [9] show that there exist learning tasks in which

[4] For example, consider a programming language $\mathcal{L}_{x,y}$ that given a command y outputs x.

memorization is only necessary for *efficient* learners. While large memorization (i.e., mutual information) implies that the algorithm is not differentially private, it does not imply the existence of an efficient attack. Indeed, except for [9, 1], these works are typically not constructive (i.e., they do not provide an efficient way to translate the memorization into an efficient privacy attack).

Computational Differential Privacy

[5, 48] considered a computational relaxation of (the standard, information-theoretic) differential privacy, where they require that the outputs of two executions on neighboring datasets are indistinguishable only from the eyes of an *efficient* observer. This relaxation turns out to be necessary for fundamental distributed tasks [47, 31] but also for some (artificial) centralized ones [8, 27]. We note that an efficient privacy attack (e.g., membership inference or reconstruction) implies that the algorithm is not computationally differentially private. But in the opposite direction, if an algorithm is not computationally differentially private, the privacy attack, while being efficient, could be very negligible (e.g., the attack might reveal only a single, insignificant, bit of information about a training example, and still violating computational differential privacy).

Membership Inference

In *membership inference (MI)* [53, 60], an adversary is given a target example and its goal is to infer whether it was included in a model's training set. Most techniques follow a paradigm of measuring some correlation/loss function between the target example and the model and checking if it exceeds some threshold. For example, [22] showed that any algorithm that given $x_1, \ldots, x_n \in \{-1, 1\}^d$ outputs $y \in [-1, 1]^d$ that is sufficiently close to the average $\frac{1}{n}\sum_{i=1}^{n} x_i$, is exposed to the MI attack that given a target example x, decides "IN" or "OUT" based on whether $\langle x, y\rangle$ (the inner-product between x and y) exceeds some threshold. The MI literature is very rich, and includes works that perform attacks on fundamental learning tasks (e.g., [52, 22, 50, 2, 1]) along with MI attacks on more complex models (e.g., [54, 60, 34, 10, 44, 50, 58]). We remark that while MI attacks are indeed a reasonable privacy concern in many scenarios, they are weaker than reconstruction attacks because the attacker needs to know the target training example beforehand while without it there might be no way to extract sensitive information.

Reconstruction

The results of [19] provided a foundation for rigorously quantifying reconstruction bounds for general query release mechanisms. In their setting, the dataset is binary, and they show that given a few statistical queries, it is possible to reveal 90% of the dataset. [21, 23, 31] provide improved results in similar settings like [19] but under weaker accuracy assumptions. In more general settings, [29] demonstrated that both Rényi Differential Privacy (DP) and Fisher information offer robust semantic assurances against reconstruction attacks. [4] introduced the concept of reconstruction robustness (ReRo), establishing a link between reconstruction attacks and DP. [32] extended [4]'s to analyze reconstruction attacks against DP-SGD. [35] furthered this research by examining the connection between the hypothesis testing interpretation of DP (specifically f-DP) and reconstruction robustness. From a more practical standpoint, there is a rich literature on reconstruction attacks on various real-world models (e.g., [26, 12, 59, 33, 13, 61, 11, 30]). In Section A (recognizing reconstruction), we focus on three of them in order to illustrate the expressiveness of our Definition 13.

Formalizing legal concepts of privacy

Some of the work towards formalizing legal privacy concepts mathematically has taken an approach similar to our "sandwiching" of the concept of reconstructing rather than attempting a precise answer to Question 1. These works did not attempt to model a legal concept exactly but rather define requirements which are clearly stricter or clearly weaker than the those of the legal concept, yet non-trivial so they can substantiate a claim that the use of a certain technology satisfies or does not satisfy the legal requirement. As one example, in their modeling of the FERPA privacy requirement,[5] [49] provided a definition which is stronger than the legal requirement, yet satisfiable by the use of differential privacy, hence providing strong evidence that the use of differential privacy (with appropriate parameters) satisfies the legal requirements. As another example is the modeling of the GDPR requirement of protection against singling out [16] defined *protection against predicate singling out*, a concept which is weaker than the legal requirement. Showing that k-anonymity does not protect against predicate singling out they hence claimed that it does neither satisfy the legal requirement.

2 Expressiveness of Narcissus resiliency

In the introduction we presented the definition of Narcissus resiliency (Definition 6) with the interpretation of "security against reconstruction attacks", where the function R encapsulates what a "valid reconstruction" means. We now show that with different instantiations of the function R, the same framework can be used to capture many other existing security notions. In this section we show this for the notions of *membership inference attacks* and for *predicate singling out*. In the full version of this paper [17], we extend this to additional security notions, including *differential privacy*, security of *one-way functions*, and security of *encryption schemes*. The fact that Narcissus resiliency is expressive enough to capture all these (seemingly unrelated) well-established notions enhances our confidence in its application as a security notion against reconstruction attacks.

2.1 Membership inference as Narcissus resiliency

Membership Inference (MI) attacks [53, 60] are a family of attacks which are applied mostly to machine learning models. The goal of an MI attack is to determine whether a given data record was part of the training data underlying the model or not. More specifically, let $\mathcal{D}$ be a distribution over data records, and let $\mathcal{M}$ be an algorithm for analyzing datasets of size n. Algorithm $\mathcal{M}$ is said to be MI-secure if no adversary $\mathcal{A}$ has a significant advantage in distinguishing between the outputs of the following two experiments:

- Sample $S \leftarrow \mathcal{D}^n$ and $z \leftarrow \mathcal{D}$ independently. Let $y \leftarrow \mathcal{M}(S)$. Output (y, z).
- Sample $S \leftarrow \mathcal{D}^n$. Let z be a uniformly random element from S. Let $y \leftarrow \mathcal{M}(S)$. Output (y, z).

The formal definition is as follows.

▶ **Definition 14** (Resilience to Membership Inference, [53, 60]). *Let $\mathcal{M} : \mathcal{X}^n \to Y$ be an algorithm that operates on a dataset, and let $\mathcal{D}$ be a distribution over $\mathcal{X}$. We say that $\mathcal{M}$ is $(\delta, \mathcal{D})$-MI-secure if for every adversary $\mathcal{A}$ it holds that*

[5] FERPA – The Family Educational Rights and Privacy Act (FERPA) – is is a Federal law that protects the privacy in education records.

$$\left| \Pr_{\substack{S\leftarrow\mathcal{D}^n \\ y\leftarrow\mathcal{M}(S) \\ z\leftarrow\mathcal{D} \\ b\leftarrow\mathcal{A}(y,z)}}[b=1] - \Pr_{\substack{S\leftarrow\mathcal{D}^n \\ y\leftarrow\mathcal{M}(S) \\ z\in_{\mathrm{R}}S \\ b\leftarrow\mathcal{A}(y,z)}}[b=1] \right| \leq \delta.$$

We show that Narcissus resiliency captures the concept of membership inference. We make use of the following function:

▶ **Definition 15.** *We define R_{MI} to be a (randomized) binary function, which takes two arguments: a dataset $Z \in \mathcal{X}^n$ and a (possibly randomized) function $f : \mathcal{X} \to \{0,1\}$. Given Z, f, to compute $R_{\mathrm{MI}}(Z,f)$, sample a point $z \in Z$ and return $f(z)$.*

▶ **Theorem 16.** *Let $\mathcal{M} : \mathcal{X}^n \to Y$ be an algorithm and let $\mathcal{D}$ be a distribution over $\mathcal{X}$. Then $\mathcal{M}$ is $(\delta,\mathcal{D})$-MI-secure if and only if it is $(0,\delta,\{\mathcal{D}^n\})$-$R_{\mathrm{MI}}$-Narcissus-resilient.*

Proof. First observe that an MI-adversary gets two arguments (an outcome y and a point z) and returns a bit, while an R_{MI}-Narcissus-adversary gets only one argument (the outcome y) and returns a binary function f that takes a point and returns a bit (aiming to satisfy R_{MI}). To help bridge between these notations, given an MI-adversary $\mathcal{A}$ and an outcome y, we write $f(\cdot) \leftarrow \mathcal{A}(y,\cdot)$ to denote the binary function obtained by fixing $\mathcal{A}$'s first argument to be y. We can interpret this as an R_{MI}-Narcissus-adversary that takes one argument (the outcome y) and returns the function $f(\cdot) \leftarrow \mathcal{A}(y,\cdot)$. It follows that

$$\Pr_{\substack{S\leftarrow\mathcal{D}^n \\ y\leftarrow\mathcal{M}(S) \\ f(\cdot)\leftarrow\mathcal{A}(y,\cdot)}}[R_{\mathrm{MI}}(S,f)=1] - \Pr_{\substack{S\leftarrow\mathcal{D}^n \\ T\leftarrow\mathcal{D}^n \\ y\leftarrow\mathcal{M}(S) \\ f(\cdot)\leftarrow\mathcal{A}(y,\cdot)}}[R_{\mathrm{MI}}(T,f)=1] = \Pr_{\substack{S\leftarrow\mathcal{D}^n \\ y\leftarrow\mathcal{M}(S) \\ z\in_{\mathrm{R}}S \\ b\leftarrow\mathcal{A}(y,z)}}[b=1] - \Pr_{\substack{S\leftarrow\mathcal{D}^n \\ y\leftarrow\mathcal{M}(S) \\ z\leftarrow\mathcal{D} \\ b\leftarrow\mathcal{A}(y,z)}}[b=1].$$

Hence, if $\mathcal{M}$ is $(\delta,\mathcal{D})$-MI-secure then it is also $(0,\delta,\{\mathcal{D}^n\})$-$R_{\mathrm{MI}}$-Narcissus-resilient. In addition, if the above two differences are positive, then the equality holds also in absolute value, and hence the condition of $(0,\delta,\{\mathcal{D}^n\})$-$R_{\mathrm{MI}}$-Narcissus-resiliency implies the condition of $(\delta,\mathcal{D})$-MI-security. Otherwise, to show that the left hand side cannot be smaller than $-\delta$, let $\hat{\mathcal{A}}$ be the R_{MI}-Narcissus-adversary that runs $\mathcal{A}$ and returns the "inverted" function $\hat{f} \equiv 1 - f$. Then,

$$\begin{aligned}
&\Pr_{\substack{S\leftarrow\mathcal{D}^n \\ T\leftarrow\mathcal{D}^n \\ y\leftarrow\mathcal{M}(S) \\ f(\cdot)\leftarrow\mathcal{A}(y,\cdot)}}[R_{\mathrm{MI}}(T,f)=1] - \Pr_{\substack{S\leftarrow\mathcal{D}^n \\ y\leftarrow\mathcal{M}(S) \\ f(\cdot)\leftarrow\mathcal{A}(y,\cdot)}}[R_{\mathrm{MI}}(S,f)=1] \\
&= -\left(\Pr_{\substack{S\leftarrow\mathcal{D}^n \\ T\leftarrow\mathcal{D}^n \\ y\leftarrow\mathcal{M}(S) \\ \hat{f}(\cdot)\leftarrow\hat{\mathcal{A}}(y,\cdot)}}[R_{\mathrm{MI}}(T,\hat{f})=1] - \Pr_{\substack{S\leftarrow\mathcal{D}^n \\ y\leftarrow\mathcal{M}(S) \\ \hat{f}(\cdot)\leftarrow\hat{\mathcal{A}}(y,\cdot)}}[R_{\mathrm{MI}}(S,\hat{f})=1] \right) \\
&= \Pr_{\substack{S\leftarrow\mathcal{D}^n \\ y\leftarrow\mathcal{M}(S) \\ \hat{f}(\cdot)\leftarrow\hat{\mathcal{A}}(y,\cdot)}}[R_{\mathrm{MI}}(S,\hat{f})=1] - \Pr_{\substack{S\leftarrow\mathcal{D}^n \\ T\leftarrow\mathcal{D}^n \\ y\leftarrow\mathcal{M}(S) \\ \hat{f}(\cdot)\leftarrow\hat{\mathcal{A}}(y,\cdot)}}[R_{\mathrm{MI}}(T,\hat{f})=1] \leq \delta.
\end{aligned}$$

which follows from the resilience of $\mathcal{M}$ against $\hat{\mathcal{A}}$. ◀

2.2 Predicate singling out as Narcissus resiliency

[16] defined a type of privacy attack called *Predicate Singling Out (PSO)*, intended to mathematically formulate the legal concept of "singling out" that appears in the General Data Protection Regulation (GDPR). Concretely, let $\mathcal{D}$ be a data distribution and let $S \leftarrow \mathcal{D}^n$ be a dataset containing n iid samples from $\mathcal{D}$. Let $\mathcal{M}$ be an algorithm that operates on S and return an outcome y. A PSO adversary gets y and aims to find a predicate p that "isolates" one record in S, meaning that it evaluates to 1 on *exactly* one record in S. Note, however, that without further restrictions this is not necessarily a hard task. In particular, even without looking at y, the adversary might choose a predicate p whose expectation over $\mathcal{D}$ is $1/n$. Such a predicate would isolate a record in S with constant probability. More generally, if the expectation of p is w, then the probability that it isolates a record in a fresh dataset of size n is $n \cdot w \cdot (1-w)^{n-1}$. Thus, for the attack to be considered "significant", the adversary has to succeed in its attack with probability noticeably higher than this. The formal definition is as follows.

▶ **Definition 17** ([16]). *Let $\mathcal{M} : \mathcal{X}^n \to Y$ be an algorithm that operates on a dataset. We say that $\mathcal{M}$ is $(\varepsilon, \delta, w_{\max})$-PSO secure if for every $w \le w_{\max}$, every distribution $\mathcal{D}$ over $\mathcal{X}$, and every adversary $\mathcal{A}$ it holds that*

$$\Pr_{\substack{S\leftarrow\mathcal{D}^n \\ y\leftarrow\mathcal{M}(S) \\ p\leftarrow\mathcal{A}(y)}}\left[\sum_{x\in S} p(x) = 1 \wedge \mathbf{E}_{\mathcal{D}}[p] \le w\right] \le e^{\varepsilon} \cdot \sup_{\substack{\text{predicate } p \\ \text{s.t. } \mathbf{E}_{\mathcal{D}}[p]\le w}} \left\{ n \cdot \mathbf{E}_{\mathcal{D}}[p] \cdot (1 - \mathbf{E}_{\mathcal{D}}[p])^{n-1} \right\} + \delta.$$

Using the paradigm of Narcissus resiliency, we present an alternative definition for predicate singling out, which is simpler in that it avoids reasoning directly about the expectations of the predicates.

▶ **Definition 18** (Narcissus singling out security). *Let $\mathcal{M} : \mathcal{X}^n \to Y$ be an algorithm. Let R be the relation that takes a dataset S and a predicate p, where $R(S,p) = 1$ if and only if p evaluates to 1 on* exactly *one point in S. We say that $\mathcal{M}$ is (ε, δ)-Narcissus-singling-out-secure if for every distribution $\mathcal{D}$ over $\mathcal{X}$ and for every attacker $\mathcal{A}$ it holds that*

$$\Pr_{\substack{S\leftarrow\mathcal{D}^n \\ y\leftarrow\mathcal{M}(S) \\ p\leftarrow\mathcal{A}(y)}}[R(S,p) = 1] \le e^{\varepsilon} \cdot \Pr_{\substack{S\leftarrow\mathcal{D}^n \\ T\leftarrow\mathcal{D}^n \\ y\leftarrow\mathcal{M}(S) \\ p\leftarrow\mathcal{A}(y)}}[R(T,p) = 1] + \delta.$$

Intuitively, Definitions 17 and 18 have similar interpretations: Finding a predicate that isolates a record in S after seeing the outcome of $\mathcal{M}$ is almost as hard as doing this without seeing the outcome of $\mathcal{M}$. However, the definitions are not equivalent. We leave open the question of understanding the relationships between these two definitions.

3 Narcissus-Resiliency Prevents Non-Trivial Extraction

Let $\mathcal{M}$ be a mechanism which is applied to a dataset S to obtain an outcome y. Let $\mathcal{B}$ be an "extraction attacker" that takes the outcome y and aims to compute a short program $\mathcal{A}$ that serves as an extraction evidence for S w.r.t. a relation R and a programming language $\mathcal{L}$. We show that if $\mathcal{M}$ is Narcissus-resilient (w.r.t. an appropriate reconstruction relation), then it prevents $\mathcal{B}$ from succeeding in its attack. Formally,

▶ **Definition 19.** *Fix a parameter $q \leq 1$ controlling the desired* quality *of extraction (as in Definition 13). We define the relation R_{ext} that takes a dataset S and a pair $(y, \mathcal{A})$ where y is an outcome of $\mathcal{M}$ and $\mathcal{A}$ is a program, and returns 1 if and only if the following two conditions hold:*

1. $R(S, \mathcal{A}(y)) = 1$, *and*
2. $\frac{|\mathcal{A}|}{K_{\mathcal{L}}(\{z\colon R(S,z)=1\})} \leq 1 - q$.

▶ **Lemma 20** (Narcissus-resiliency prevents non-trivial extraction)**.** *Let $\mathcal{M}$ be an $(\varepsilon, \delta, \mathcal{F})$-$R_{\text{ext}}$-Narcissus-resilient according to Definition 6, and consider an adversary $\mathcal{B}$ that takes the outcome of $\mathcal{M}$ and outputs an $\mathcal{L}$-program $\mathcal{A}$. Then for every $\mathcal{D} \in \mathcal{F}$ we have*

$$\Pr_{\substack{S\leftarrow\mathcal{D}\\ y\leftarrow\mathcal{M}(S)\\ \mathcal{A}\leftarrow\mathcal{B}(y)}}\left[\begin{array}{c}\mathcal{A} \text{ is an } (R,\mathcal{L})\text{-extraction}\\ \text{evidence of } S \text{ from } y\end{array}\right] \leq e^{\varepsilon}\cdot \Pr_{\substack{S\leftarrow\mathcal{D}\\ T\leftarrow\mathcal{D}\\ y\leftarrow\mathcal{M}(S)\\ \mathcal{A}\leftarrow\mathcal{B}(y)}}\left[\begin{array}{c}\mathcal{A} \text{ is an } (R,\mathcal{L})\text{-extraction}\\ \text{evidence of } T \text{ from } y\end{array}\right] + \delta$$

Proof. This follows directly from the assumption that $\mathcal{M}$ is R_{ext}-Narcissus-resilient. Formally, let $\hat{\mathcal{B}}$ denote the adversary that on input y returns $(y, \mathcal{B}(y))$. Then,

$$\begin{aligned}\Pr_{\substack{S\leftarrow\mathcal{D}\\ y\leftarrow\mathcal{M}(S)\\ \mathcal{A}\leftarrow\mathcal{B}(y)}}\left[\begin{array}{c}\mathcal{A} \text{ is an } (R,\mathcal{L})\text{-extraction}\\ \text{evidence of } S \text{ from } y\end{array}\right] &= \Pr_{\substack{S\leftarrow\mathcal{D}\\ y\leftarrow\mathcal{M}(S)\\ (y,\mathcal{A})\leftarrow\hat{\mathcal{B}}(y)}}\left[R_{\text{ext}}(S,(y,\mathcal{A}))=1\right]\\ &\leq e^{\varepsilon}\cdot \Pr_{\substack{S\leftarrow\mathcal{D}\\ T\leftarrow\mathcal{D}\\ y\leftarrow\mathcal{M}(S)\\ (y,\mathcal{A})\leftarrow\hat{\mathcal{B}}(y)}}\left[R_{\text{ext}}(T,(y,\mathcal{A}))=1\right]+\delta\\ &\leq e^{\varepsilon}\cdot \Pr_{\substack{S\leftarrow\mathcal{D}\\ T\leftarrow\mathcal{D}\\ y\leftarrow\mathcal{M}(S)\\ \mathcal{A}\leftarrow\mathcal{B}(y)}}\left[\begin{array}{c}\mathcal{A} \text{ is an } (R,\mathcal{L})\text{-extraction}\\ \text{evidence of } T \text{ from } y\end{array}\right]+\delta\end{aligned}$$ ◀

A possible downside of Lemma 20 is that it leverages a *different* reconstruction relation for Narcissus-resiliency compared to the extraction relation. This was needed in the proof because our definition of extraction considered two conditions: not only that the attacker needs to satisfy the relation R, but it needs to do it with a "short enough" program. To capture this in the above lemma, we incorporated this condition in the reconstruction relation used for Narcissus-resiliency. This can be avoided in cases where for some parameter k we have:

1. The adversary $\mathcal{B}$ always outputs a program of length at most k; and
2. The underlying data distribution $\mathcal{D}$ is such that with overwhelming probability over sampling $S \leftarrow \mathcal{D}$ we have that $\frac{k}{K_{\mathcal{L}}(\{z\colon R(S,z)=1\})} \leq 1 - q$.

That is, in cases when sampling a dataset $S \leftarrow \mathcal{D}$ then with overwhelming probability the Kolmogorov complexity $K_{\mathcal{L}}(\{z\colon R(S,z)=1\})$ is high. Indeed, if this is the case, then the second condition in Definition 19 holds with overwhelming probability, and hence can be ignored, which unifies the two relations R and R_{ext}.

3.1 Limitation of Lemma 20

It is important to note that Lemma 20 is limited to attackers $\mathcal{B}$ that only see the output y of the mechanism. But some of the real-world attacks also use the training data in order to construct an extraction evidence. E.g., in the Diffusion Model attack of [11] (see Section A.3), the attacker $\mathcal{B}$ first processes the input data in order to find captions of "interesting" images,

and then define $\mathcal{A}$ as the short program that simply queries the model on one of these captions. We note that Narcissus-resiliency does not prevent such attacks in general, because $\mathcal{B}$ uses the training data to generate $\mathcal{A}$. But Lemma 20 does imply that it is impossible to achieve non-trivial extraction of training examples, without knowing them in advanced.

References

1 Idan Attias, Gintare Karolina Dziugaite, Mahdi Haghifam, Roi Livni, and Daniel M. Roy. Information complexity of stochastic convex optimization: Applications to generalization and memorization, 2024. doi:10.48550/arXiv.2402.09327.

2 Achraf Azize and Debabrota Basu. How much does each datapoint leak your privacy? quantifying the per-datum membership leakage. *CoRR*, abs/2402.10065, 2024. doi:10.48550/arXiv.2402.10065.

3 Marshall Ball, Yanyi Liu, Noam Mazor, and Rafael Pass. Kolmogorov comes to cryptomania: On interactive kolmogorov complexity and key-agreement. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023*, pages 458–483, 2023. doi:10.1109/FOCS57990.2023.00034.

4 Borja Balle, Giovanni Cherubin, and Jamie Hayes. Reconstructing training data with informed adversaries. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1138–1156. IEEE, 2022. doi:10.1109/SP46214.2022.9833677.

5 Amos Beimel, Kobbi Nissim, and Eran Omri. Distributed private data analysis: Simultaneously solving how and what. In *Annual International Cryptology Conference (CRYPTO)*, pages 451–468, 2008. doi:10.1007/978-3-540-85174-5_25.

6 Gavin Brown, Mark Bun, Vitaly Feldman, Adam Smith, and Kunal Talwar. When is memorization of irrelevant training data necessary for high-accuracy learning? In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 123–132, 2021. doi:10.1145/3406325.3451131.

7 Gavin Brown, Mark Bun, and Adam Smith. Strong memory lower bounds for learning natural models. In *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 4989–5029, 2022. URL: https://proceedings.mlr.press/v178/brown22a.html.

8 Mark Bun, Yi-Hsiu Chen, and Salil P. Vadhan. Separating computational and statistical differential privacy in the client-server model. In *Theory of Cryptography - 14th International Conference, TCC 2016-B*, volume 9985, pages 607–634, 2016. doi:10.1007/978-3-662-53641-4_23.

9 Mark Bun and Mark Zhandry. Order-revealing encryption and the hardness of private learning. In *TCC 2016-A*, volume 9562, pages 176–206, 2016. doi:10.1007/978-3-662-49096-9_8.

10 Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914, 2022. doi:10.1109/SP46214.2022.9833649.

11 Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *Proceedings of the 32nd USENIX Conference on Security Symposium*, SEC '23, 2023.

12 Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: evaluating and testing unintended memorization in neural networks. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, pages 267–284, USA, 2019. USENIX Association. URL: https://www.usenix.org/conference/usenixsecurity19/presentation/carlini.

13 Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium, USENIX Security 2019*, pages 267–284, 2019. URL: https://www.usenix.org/conference/usenixsecurity19/presentation/carlini.

14 Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

15 Gregory J. Chaitin. On the simplicity and speed of programs for computing infinite sets of natural numbers. *J. ACM*, 16(3):407–422, 1969. doi:10.1145/321526.321530.

16 Aloni Cohen and Kobbi Nissim. Towards formalizing the gdpr's notion of singling out. *Proc. Natl. Acad. Sci. USA*, 117(15):8344–8352, 2020. doi:10.1073/PNAS.1914598117.

17 Edith Cohen, Haim Kaplan, Yishay Mansour, Shay Moran, Kobbi Nissim, Uri Stemmer, and Eliad Tsfadia. Data reconstruction: When you see it and when you don't, 2024. doi:10.48550/arXiv.2405.15753.

18 Rachel Cummings, Shlomi Hod, Jayshree Sarathy, and Marika Swanberg. Attaxonomy: Unpacking differential privacy guarantees against practical adversaries, 2024. doi:10.48550/arXiv.2405.01716.

19 Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *PODS*, pages 202–210. ACM, 2003. doi:10.1145/773153.773173.

20 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, volume 3876, pages 265–284, 2006. doi:10.1007/11681878_14.

21 Cynthia Dwork, Frank McSherry, and Kunal Talwar. The price of privacy and the limits of lp decoding. In *STOC*, pages 85–94. ACM, 2007. doi:10.1145/1250790.1250804.

22 Cynthia Dwork, Adam D. Smith, Thomas Steinke, Jonathan R. Ullman, and Salil P. Vadhan. Robust traceability from trace amounts. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015*, pages 650–669, 2015. doi:10.1109/FOCS.2015.46.

23 Cynthia Dwork and Sergey Yekhanin. New efficient attacks on statistical disclosure control mechanisms. In *CRYPTO*, pages 469–480. Springer, 2008. doi:10.1007/978-3-540-85174-5_26.

24 Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pages 954–959, 2020. doi:10.1145/3357713.3384290.

25 Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: discovering the long tail via influence estimation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, 2020.

26 Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 1322–1333, 2015. doi:10.1145/2810103.2813677.

27 B. Ghazi, R. Ilango, P. Kamath, R. Kumar, and P. Manurangsi. Towards separating computational and statistical differential privacy. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 580–599, 2023.

28 Peter D. Grünwald and Paul M. B. Vitányi. Kolmogorov complexity and information theory. with an interpretation in terms of questions and answers. *Journal of Logic, Language and Information*, 12(4):497–529, 2003. doi:10.1023/A:1025011119492.

29 Chuan Guo, Brian Karrer, Kamalika Chaudhuri, and Laurens van der Maaten. Bounding training data reconstruction in private (deep) learning. In *International Conference on Machine Learning, ICML 2022*, volume 162, pages 8056–8071, 2022. URL: https://proceedings.mlr.press/v162/guo22c.html.

30 Niv Haim, Gal Vardi, Gilad Yehudai, Ohad Shamir, and Michal Irani. Reconstructing training data from trained neural networks. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems, NeurIPS 2022*, 2022.

31 Iftach Haitner, Noam Mazor, Jad Silbak, and Eliad Tsfadia. On the complexity of two-party differential privacy. In *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1392–1405. ACM, 2022. doi:10.1145/3519935.3519982.

32 Jamie Hayes, Borja Balle, and Saeed Mahloujifar. Bounding training data reconstruction in DP-SGD. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

33 Zecheng He, Tianwei Zhang, and Ruby B. Lee. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference*, ACSAC '19, pages 148–162, 2019. doi:10.1145/3359789.3359824.

34 Matthew Jagielski, Om Thakkar, Florian Tramer, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Guha Thakurta, Nicolas Papernot, and Chiyuan Zhang. Measuring forgetting of memorized training examples. In *The Eleventh International Conference on Learning Representations*, 2023.

35 Georgios Kaissis, Jamie Hayes, Alexander Ziller, and Daniel Rueckert. Bounding data reconstruction attacks with the hypothesis testing interpretation of differential privacy, 2023. doi:10.48550/arXiv.2307.03928.

36 K.-I. Ko. On the notion of infinite pseudorandom sequences. *Theor. Comput. Sci.*, 48(1):9–33, 1986. doi:10.1016/0304-3975(86)90081-2.

37 A. N. Kolmogorov. Three approaches to the quantitative definition of information. *International Journal of Computer Mathematics*, 2(1-4):157–168, 1968.

38 L Levin. Universal search problems (russian), translated to english by trakhtenbrot (1984). *Problems of Information Transmission*, 9(3):265–266, 1973.

39 Yanyi Liu and Rafael Pass. On one-way functions and kolmogorov complexity. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020*, pages 1243–1254, 2020. doi:10.1109/FOCS46700.2020.00118.

40 Yanyi Liu and Rafael Pass. On the possibility of basing cryptography on exp$\neq$ BPP. In *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021*, volume 12825, pages 11–40, 2021. doi:10.1007/978-3-030-84242-0_2.

41 Yanyi Liu and Rafael Pass. Characterizing derandomization through hardness of levin-kolmogorov complexity. In Shachar Lovett, editor, *37th Computational Complexity Conference, CCC 2022*, volume 234, pages 35:1–35:17, 2022. doi:10.4230/LIPICS.CCC.2022.35.

42 Yanyi Liu and Rafael Pass. On one-way functions from np-complete problems. In *37th Computational Complexity Conference, CCC 2022*, volume 234, pages 36:1–36:24, 2022. doi:10.4230/LIPICS.CCC.2022.36.

43 Roi Livni. Information theoretic lower bounds for information theoretic upper bounds. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*, 2023.

44 Yunhui Long, Lei Wang, Diyue Bu, Vincent Bindschaedler, Xiaofeng Wang, Haixu Tang, Carl A. Gunter, and Kai Chen. A pragmatic approach to membership inferences on machine learning models. In *2020 IEEE European Symposium on Security and Privacy*, pages 521–534, 2020. doi:10.1109/EUROSP48549.2020.00040.

45 Luc Longpré and Sarah Mocas. Symmetry of information and one-way functions. *Information Processing Letters*, 46(2):95–100, 1993. doi:10.1016/0020-0190(93)90204-M.

46 Jean loup Gailly and Mark Adle. zlib compression library, 2004.

47 Andrew McGregor, Ilya Mironov, Toniann Pitassi, Omer Reingold, Kunal Talwar, and Salil Vadhan. The limits of two-party differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 81–90, 2010. doi:10.1109/FOCS.2010.14.

48 Ilya Mironov, Omkant Pandey, Omer Reingold, and Salil Vadhan. Computational differential privacy. In *Annual International Cryptology Conference (CRYPTO)*, pages 126–142, 2009. doi:10.1007/978-3-642-03356-8_8.

49 Kobbi Nissim, Aaron Bembenek, Alexandra Wood, Mark Bun, Marco Gaboardi, Urs Gasser, David R. O'Brien, Thomas Steinke, and Salil Vadhan. Bridging the gap between computer science and legal approaches to privacy. *Harvard Journal of Law & Technology*, 31(2):687–780, 2018.

50 Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, 2019.

51 Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems, NeurIPS 2022*, 2022.

52 Sriram Sankararaman, Guillaume Obozinski, Michael I Jordan, and Eran Halperin. Genomic privacy and limits of individual detection in a pool. *Nature genetics*, 41(9):965–967, 2009.

53 Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017. doi:10.1109/SP.2017.41.

54 Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy, SP 2017*, pages 3–18, 2017. doi:10.1109/SP.2017.41.

55 Michael Sipser. A complexity theoretic approach to randomness. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, STOC '83, pages 330–335, 1983. doi:10.1145/800061.808762.

56 R.J. Solomonoff. A formal theory of inductive inference. *Information and Control*, 7(1):1–22, 1967.

57 B.A. Trakhtenbrot. A survey of russian approaches to perebor (brute-force searches) algorithms. *Annals of the History of Computing*, 6(4):384–400, 1984. doi:10.1109/MAHC.1984.10036.

58 Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. On the importance of difficulty calibration in membership inference attacks. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022. URL: https://openreview.net/forum?id=3eIrli0TwQ.

59 Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. Neural network inversion in adversarial setting via background knowledge alignment. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, pages 225–240. Association for Computing Machinery, 2019. doi:10.1145/3319535.3354261.

60 Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018. doi:10.1109/CSF.2018.00027.

61 H. Yin, P. Molchanov, J. M. Alvarez, Z. Li, A. Mallya, D. Hoiem, N. K. Jha, and J. Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8712–8721. IEEE Computer Society, 2020.

62 A Zvonkin and L Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Mathematical Surveys*, 6:83–124, 1970.

A Recognizing Extraction

Following the discussion in Section 1.2, we use the terminology of "extraction" rather than "reconstruction" in the setting where the dataset S and the model y are fixed. We formally define how to recognize a valid extraction attack on a given (fixed) model y (as described in Section 1.2), use the definition to explain the validity of some real-world attacks, and prove that in general the problem of verifying a valid attack is computationally hard assuming that cryptography exists.

A.1 Defining Extraction via Kolmogorov Complexity

In this paper, we think of extraction as a randomized process that can fail with some probability, so we use the following variant of Definition 12.

▶ **Definition 21** ($K_{\mathcal{L}}$-Complexity, a probabilistic version of Definition 12)**.** *Let $\mathcal{L}$ be a programming language. The $K_{\mathcal{L}}$-complexity of a string x, denoted by $K_{\mathcal{L}}(x)$, is the length of the shortest $\mathcal{L}$-program that with probability $2/3$ outputs the string x and halts.*

In theoretical results, this complexity is usually defined w.r.t. a fixed *Universal Turing Machine* $\mathcal{U}$ that gets a description $\mathcal{M}$ of a Turing machine and outputs $\mathcal{U}(\mathcal{M})$. Then, $K(x) = K_{\mathcal{U}}(x)$ is defined by the shortest $\mathcal{M}$ such that $\Pr[\mathcal{U}(\mathcal{M}) = x] \geq 2/3$. This is good enough for *asymptotic* analysis since for any programming language $\mathcal{L}$ and every x it holds that $K(x) \leq K_{\mathcal{L}}(x) + O(1)$, where $O(1)$ denotes a constant that is independent of $|x|$.[6] However, when we would like to deal with concrete quantities (where constants matter), we must explicitly specify a concrete programming language $\mathcal{L}$, and therefore we do not omit it from the notation.

After fixing the programming language $\mathcal{L}$, the quantity $K_{\mathcal{L}}(x)$ can be thought of as the analogous to the "entropy" of the string x. As with entropy, we can also consider a conditional version of the K-complexity (the analog of conditional entropy) as well as "mutual K-information" (the analog of mutual information) [62, 38, 57, 45, 28, 41, 3]:

▶ **Definition 22** (Conditional $K_{\mathcal{L}}$-Complexity, and mutual $K_{\mathcal{L}}$-information)**.** *Given a programming language $\mathcal{L}$ and two strings x, y, we define the* conditional $K_{\mathcal{L}}$-complexity of x given y*, denoted by $K_{\mathcal{L}}(x \mid y)$, as the length of the shortest program in $\mathcal{L}$ that given y as input, with probability $2/3$ outputs x and halts. We define the* mutual $K_{\mathcal{L}}$-information *of x, y as $KI_{\mathcal{L}}(x; y) = K_{\mathcal{L}}(x) - K_{\mathcal{L}}(x \mid y)$.*

Intuitively, $KI_{\mathcal{L}}(x; y)$ is high if it is possible to extract x from y using a short $\mathcal{L}$-program, but it is not possible to do so without y. In our context of extracting an input example x from a model y, we should also allow to output elements z that are "similar" to x (which are considered as a valid extraction of x under some metric), or perhaps the goal is to output a list of elements that some of them are close to training examples. Therefore, we would be interested in an extension of the K-complexity into a set of strings which allows much more flexibility.

▶ **Definition 23** (Extension of Definitions 21 and 22 for sets)**.** *Let $\mathcal{L}$ be a programming language. The $K_{\mathcal{L}}$-complexity of a set of strings X, denoted by $K_{\mathcal{L}}(X)$, is the length of the shortest $\mathcal{L}$-program that with probability $2/3$, outputs a string $x \in X$ and halts. Similarly to Definition 22, we define $K_{\mathcal{L}}(X \mid y)$ as the length of the shortest $\mathcal{L}$-program that given an input y, outputs w.p. $2/3$ a string $x \in X$ and halts, and we define $KI_{\mathcal{L}}(X; y) = K_{\mathcal{L}}(X) - K_{\mathcal{L}}(X \mid y)$.*

Note that by definition, $KI_{\mathcal{L}}(X; y) = \min_{x \in X} K_{\mathcal{L}}(x) - \min_{x \in X} K_{\mathcal{L}}(x \mid y)$, and these minimums are not necessarily realized by the same x.

A few examples are in order. In all the following examples, we use $\mathcal{L}$ as a fixed Universal Turing Machine language, and for $x = (x^1, \ldots, x^d), z = (z^1, \ldots, z^d) \in \{0,1\}^d$ we denote by $\mathsf{NHamDist}(x, z) = \frac{1}{d} \cdot \left|\{i \in [n] \colon x^i \neq z^i\}\right|$ the *Normalized Hamming Distance* between x and z.

---

[6] Let $\mathcal{C}$ be a $\mathcal{U}$-program (i.e., a Turing machine description) that serves as a compiler from a language $\mathcal{L}$ to $\mathcal{U}$, i.e., give a $\mathcal{L}$-program $\mathcal{M}$, $C(\mathcal{M})$ outputs a $\mathcal{U}$-program $\mathcal{M}'$ that acts as $\mathcal{M}$. So for every x, if it can be described using an $\mathcal{L}$-program $\mathcal{M}$, then it can be described using a $\mathcal{U}$-program $\mathcal{C}(\mathcal{M})$ of length $|\mathcal{M}| + |\mathcal{C}| = |\mathcal{M}| + O(1)$ (note that $|\mathcal{C}|$ is independent of $|x|$).

▶ **Example 24.** For $x \in \{0,1\}^d$ define $R_x = \{(z_1, z_2) \in \{0,1\}^{2d} : \min_{i\in\{1,2\}} \mathsf{NHamDist}(x, z_i) \leq 1/2\}$, (i.e., in this example, R_x defines the task of outputting two strings that at least one of them agrees with x on at least half of the bits). Note that $K_{\mathcal{L}}(R_x) = O(\log d)$ for any x, because the program that outputs a uniformly random $(z_1, z_2) \leftarrow \{0,1\}^{2d}$ – e.g., a for loop executing for $2d$ iterations and outputing a random bit in each – satisfies $(z_1, z_2) \in R_x$ with probability $3/4$.

▶ **Example 25.** For $x \in \{0,1\}^d$ define R_x as the set of all the tuples of the form $z = (z_1, \ldots, z_m)$, for $m \leq \text{poly}(d)$ (for some fixed polynomial), such that $\min_{i\in[m]} \mathsf{NHamDist}(x, z_i) \leq 1/4$ (i.e., at least one of the z_i's agrees with $3/4$ of the bits of x). For a uniformly random $x \leftarrow \{0,1\}^d$, it holds with high probability that $K_{\mathcal{L}}(R_x) \geq (1 - o(1))d/2$ (i.e., for most strings x, any algorithm that aims to output a string that agrees with $3/4$ of the bits of x, essentially must memorize $\approx d/2$ bits of information about x). Indeed, given $y = (x^1, \ldots, x^{d/2})$ (the first half of the bits of x) as input, we have $K(R_x \mid y) = O(\log d)$ because the algorithm that chooses $m = 2$ uniformly random strings $(z_1, z_2) \leftarrow \{0,1\}^{d/2}$ and outputs $z = (y \circ z_1,\ y \circ z_2)$ satisfies $z \in R_x$ with probability $3/4$.

▶ **Example 26.** Let $S = (x_1, \ldots, x_n) \in (\{0,1\}^d)^n$, and define R_S as the set of all the tuples $z = (z_1, \ldots, z_m)$ such that there exist $i \in [n]$ and $j \in [m]$ with $\mathsf{NHamDist}(x_i, z_j) \leq 1/4$. Assume $n, m \leq \text{poly}(d)$ (for some fixed polynomial). Then as in Example 25, it can be shown that when S consists of uniformly random strings, then $K_{\mathcal{L}}(R_S) \geq (1 - o(1))d/2$ with high probability, meaning that solving this problem requires to memorize at least $\approx d/2$ bit of information from one of the strings.

We are now ready to reformulate the extraction definitions of [14, 11].

▶ **Definition 27** (Extractable Information)**.** *Let S be a set of elements over a domain $\mathcal{X}$, let $R : \mathcal{X}^* \times \{0,1\}^* \to \{0,1\}$ be an extraction relation (i.e., $R(S, z) = 1$ means that z is considered a valid extraction of elements in S), let $\mathcal{L}$ be a programming language, and let $R_S = \{z : R(S, z) = 1\}$. We say that a model y contains $(R, \mathcal{L}, \tau)$-extractable information about S if*

$$KI_{\mathcal{L}}(R_S; y) \geq \tau.$$

In order to prove that S is extractable from a model y, it suffices to present an extractor:

▶ **Definition 28** (Extractor, redefinition of Definition 13)**.** *Let $R : \mathcal{X}^* \times \{0,1\}^* \to \{0,1\}$ be an extraction relation and denote $R_S = \{z : R(S, z) = 1\}$. Let $q \in [0, 1)$ be a quality parameter. We say that an $\mathcal{L}$-program $\mathcal{A}$ is an $(\mathcal{L}, R, q)$-extractor of $S \in \mathcal{X}^*$ from y if the following holds:*

1. $\Pr[\mathcal{A}(y) \in R_S] \geq 2/3$*, and*
2. $K_{\mathcal{L}}(R_S) \geq \frac{|\mathcal{A}|}{1-q}$.

Note that when q is closer to 1 is means a more significant extraction.

▷ Claim 29. If there exists an $(\mathcal{L}, R, q)$-extractor $\mathcal{A}$ of S from y, then y contains $(\mathcal{L}, R, \tau = \frac{q}{1-q} \cdot |\mathcal{A}|)$-extractable information about S (Definition 27).

Proof. Compute

$$KI_{\mathcal{L}}(R_S; y) = K_{\mathcal{L}}(R_S) - K_{\mathcal{L}}(R_S \mid y) \geq \frac{|\mathcal{A}|}{1-q} - |\mathcal{A}| = \frac{q}{1-q} \cdot |\mathcal{A}|. \qquad \triangleleft$$

A.2 Capturing Interesting Extractions using the Predicate

In the previous examples (e.g., Example 26), we considered simple relations R that only check for a small distance from an example point. But in the real-world, when thinking about S as the training dataset and on y as the trained model (over S), it is very likely that S will contain duplicated data, and in such cases, it might be considered ok to memorize such data. As a concrete example, if we think about a chat-bot model, then we can query it with "Print the Bible" and it will likely do that. I.e., even though it is indeed a valid extraction (as the K-complexity of the Bible is large), this is not an interesting one since the Bible content is probably duplicated in many training examples. In order to define more interesting relations R, it is reasonable to consider extraction of only *k-Eidetic Memorizated* strings x ([14]) for some small threshold k, which means that such x's only appear at most k times in S.

A.3 Real-World Examples

In order to illustrate the generality of our definitions, we next pick three different types of real-world attacks, and explain what is happening in each of them in terms of Definition 28.

Extraction from Large Language Models [14]

[14] present extraction attacks on GPT-2 by essentially generating many samples from GPT-2 when the model is conditioned on (potentially empty) prefixes. Then they sort each generation according to some metric to remove duplications, and this gives a set of potentially memorized texts from training examples. In one of their methods, they choose the top-m interesting ones by assigning a score to each such text which combines the compressibility of the text using some compression algorithm like zlib [46], and the likelihood of it, e.g., using a "perplexity" measure [13] (higher compressibility and likelihood increase the score). They show that noticeable fraction of such strings are "Eidetic Memorizated", meaning that they only appear a small number of times in the training data (which makes them more interesting).

In terms of our extraction methodology, the training dataset $S = (x_1, \ldots, x_n)$ consists of webpages, and an *interesting extractable text* is a string s that: (1) has high entropy (i.e., high K-complexity, which can be estimated using heuristics like zlib compression) and (2) appears only small number of times in the training examples (here, "appear" in webpage x_i means $s \subseteq x_i$). The attack outputs a list of strings $z = (z_1, \ldots, z_m)$, and succeed when there exists a sub-list $(z_{j_1}, \ldots, z_{j_\ell})$ such that each string z_{j_i} is interesting and extractable, and the K-complexity of the entire list is higher than some threshold (which, again, can be estimated using heuristics like zlib compression).

When the K-complexity of the resulting sub-list is much higher than the length of the attacking code that extracted this information from the GPT-2 model, the attack is considered to have high quality.

Extraction from Diffusion Models [11]

[11] present several extraction attacks on Diffusion models, where all the attacks assume that the training data is given in advance. As one concrete example, they managed to extract training images from Imagen ([51]), a 2 billion parameter text-to-image diffusion model, by first searching for the captions of "outliers" in the training data (that is, training images that are less similar to other training images according to some metric), and then showed that when querying the model on such captions, some of them result with a good approximation of unique training images.

In terms of our extraction methodology, the end program that queries the model on a specific caption that leads to extraction of a unique training image is extremely short compared to the K-complexity of the image that was extracted. So this should be considered a very high-quality extraction to a predicate of the following type (for some threshold parameters $T > t$):

$$R(S = (x_1, \ldots, x_n), z) = 1 \iff \exists i \in [n] \text{ s.t. } (\ell_2(x_i, z) < t \text{ and } \forall j \in [n] \setminus \{i\} : \ell_2(x_i, x_j) > T).$$

Extraction from Binary Classifiers trained by Large Neural Networks [30]

[30] present a general attack that works on large class of binary classifiers trained by Neural networks with many parameters p. Roughly, their attack works as follows: Given the parameters of the model $y \in \mathbb{R}^p$, they define an optimization problem (that depends on y) on variables $z_1, \ldots, z_m \in \mathbb{R}^d$ and $\lambda_1, \ldots, \lambda_m \in \mathbb{R}$ and optimize it, and show that in the regime $p > n \cdot d$, the solution $z_1, \ldots, z_m$ is likely to contain a good approximation of many data points.

As one example of their experimental evaluation, they trained the model on the training images of CIFAR10 on vehicles vs. animals classification, and showed that their optimization program, which is defined based on the model $y \in \mathbb{R}^p$ given as input, managed to extract a good approximation of many training images (the top 45 extractions are presented in their paper).

In terms of our extraction methodology, given the training data $S = (x_1, \ldots, x_n)$, we can define the extraction relation

$$R(S = (x_1, \ldots, x_n), z = (z_1, \ldots, z_m)) = 1 \iff |\{i \in [n] \colon \exists j \in [m] \text{ s.t. } z_j \text{ is "close" to } x_i\}| \geq t,$$

for some parameter t (say, $t = 45$), where "close" is according to some metric. Under the reasonable assumption that $K(R_S)$ is much higher than the length of their attack (which is given the model y as input), this is considered a high quality extraction.

A.4 Inherent Limitations of Definition 28

The main limitation of our Definition 28 is that Item 2 (lower-bounding the Kolmogorov Complexity) is not verifiable, as computing the K-complexity is an intractable problem. While in practice the K-complexity can be estimated using Heuristics like zlib compression [46], any such Heuristic can fail to determine some highly compressible patterns, and as we demonstrate next in Section A.4.1, this problem is indeed inherent assuming that basic cryptography exists.

A well studied tractable variant of the K-complexity is the $t(\cdot)$-time bounded K-complexity, denoted by K^t-complexity, where $K^t(x)$ is the length of the shortest program that outputs x within $t(|x|)$ steps ([37, 55, 57, 36]). However, there is no efficient algorithm for computing the K^t-complexity.[7] Very recently, in a sequence of breakthrough results in cryptography and complexity theory [39, 40, 42, 41], it was shown that (a close variant of) this problem is hard if and only if one-way functions, the most basic cryptographic primitives, exist. Since a real-world verifier is a polynomial-time algorithm, this means that switching from

[7] As surveyed by [57], the problem of efficiently determining the K^t-complexity for $t(n) = \text{poly}(n)$ predates the theory of NP-completeness and was studied in the Soviet Union since the 60s as a candidate for a problem that requires "brute-force search" (see Task 5 on page 392 in [57]).

$K(\cdot)$ to the $K^{\mathrm{poly}(n)}(\cdot)$ variant will essentially not make any difference. Actually, the hardness of verification is inherent, and in the next two sections we show, under common cryptographic assumptions, how to construct triplets $(\mathcal{A}, x, y)$, with efficient $\mathcal{A}$, such that it is computationally hard to verify whether $\mathcal{A}$ extracts information about x from y.

A.4.1 Hardness of Verification

Our harness example is based on the existence of Pseudorandom Generator (PRG).

▶ **Definition 30** (Pseudorandom Generator (PRG)). *A PRG is a* PPT *algorithm G that maps d bits to $\ell(d)$ bits (for $\ell(d) > d$) such that for every* PPT *algorithm V and every $d \in \mathbb{N}$ it holds that*

$$\Pr[V(G(\mathcal{U}_d)) = 1] \leq \Pr\left[V(\mathcal{U}_{d+\ell(d)}) = 1\right] + \mathrm{neg}(d),$$

where $\mathcal{U}_m$ denotes the uniform distribution over $\{0,1\}^m$.

▶ **Lemma 31.** *Assume there exists a PRG G that maps d bits to d^4 bits. Let $\mathcal{L}$ be a Universal Turing Machine programming language with oracle access to G. Then there exists a* PPT *algorithm* Attacker *such that on input 1^d and $b \in \{0,1\}$, outputs a deterministic program $\mathcal{A}$ and two strings $x, y \in \{0,1\}^{d^4}$ such that the following holds w.r.t. a random execution $(\mathcal{A}, x, y) \leftarrow \mathsf{Attacker}(1^d, b)$:*

1. *$|\mathcal{A}| = d^2 + O(1)$ and $\mathcal{A}(y) = x$ (which is an evidence for $K_{\mathcal{L}}(x \mid y) \leq d^2 + O(1)$).*
2. *If $b = 0$ then $K_{\mathcal{L}}(x) \leq d + O(1)$ (and in particular, there is a deterministic $\mathrm{poly}(d)$-time $(d + O(1))$-size algorithm that outputs x).*
3. *If $b = 1$ then x is chosen uniformly over $\{0,1\}^{d^4}$ (which yields $K_{\mathcal{L}}(x) \geq (1 - o(1))d^4$ w.h.p.),*
4. *For every* PPT *$\mathcal{V}$ it holds that*

$$\Pr_{b \leftarrow \{0,1\},\, (\mathcal{A},x,y) \leftarrow \mathsf{Attacker}(1^d,b)}[\mathcal{V}(\mathcal{A}, x, y) = b] \leq 1/2 + \mathrm{neg}(d).$$

Note that when $b = 1$, $\mathcal{A}$ extracts most of the information about x from y (quality $1 - \frac{1+o(1)}{d^2}$), while the case $b = 0$ is clearly not an extraction as $K_{\mathcal{L}}(x) \ll |\mathcal{A}|$. Yet, by Item 4, any PPT verifier $\mathcal{V}$ who only sees $\mathcal{A}, x, y$ cannot distinguish between the two cases.

Proof. The proof holds by considering the following Attacker algorithm.

Algorithm 1 Attacker.

- Input: 1^d and $b \in \{0,1\}$.
- Oracle: PRG G: $\{0,1\}^d \to \{0,1\}^{d^4}$.
- Operation:
  1. If $b = 0$: Sample $s \leftarrow \{0,1\}^d$ and let $x = (x_1, \ldots, x_{d^4}) = G(s)$.
  2. Else (i.e., $b = 1$): Sample $x = (x_1, \ldots, x_{d^4}) \leftarrow \{0,1\}^{d^4}$.
  3. Let $y = (x_1, \ldots, x_{d^4 - d^2})$.
  4. Let $\mathcal{A}$ be the $(d^2 + O(1))$-size algorithm that has $x' = (x_{d^4-d^2+1}, \ldots, x_{d^4})$ hard-coded, and given y as input, outputs $y \circ x'$ and halts.
  5. Output $(\mathcal{A}, x, y)$.

Note that Items 1–3 of the lemma holds by construction. Furthermore, Item 4 immediately holds since G is a PRG. ◀

In the above example, we defined a hardness problem w.r.t. the predicate $R(x, z) = 1 \iff z = x$. But it can be easily be relaxed to a more realistic predicate of the form

$$R(S = (x_1, \ldots, x_n), z = (z_1, \ldots, z_m)) = 1 \iff \exists i \in [n], j \in [m] \text{ s.t. } \{\mathsf{NHamDist}(x_i, z_j)\} \leq 1/2 - \alpha,$$

for any fixing of a constant $\alpha > 0$ and $n, m \leq \text{poly}(d)$ (i.e., when the task is to output a list of strings that at least one of them has non-trivial agreement with one of the elements of S).[8]

[8] The idea is to replace Step 3 of Attacker with $y = (x_1, \ldots, x_{\alpha d^4 - d^2 \log d})$, where x is one of the elements in $\mathcal{S}$, and in Step 4 to define $\mathcal{A}$ as the $O(d^2 \log d)$-size program that has $x' = (x_{\alpha d^4 - d^2 \log d}, \ldots, x_{\alpha d^4 + d^2 \log d})$ hard-coded, and given y as input, samples $x'' \leftarrow \{0,1\}^{(1-\alpha)d^4 - d^2 \log d}$ and outputs $z = y \circ x' \circ x''$. By concentration bounds of the binomial distribution, it holds that $\Pr[\mathsf{NHamDist}(x, z) \leq 1/2 - \alpha] \geq 1 - \text{neg}(d)$. But without memorizing x' or at least $\Omega(d^2 \log d)$ bits of information about one $x \in \mathcal{S}$, the probability that a random guess of the bits after y would lead to $1/2 - \alpha$ Normalized Hamming distance from an element $x \in \mathcal{S}$ is $\text{neg}(d)$.

Derandomized Squaring: An Analytical Insight into Its True Behavior

Gil Cohen
Tel Aviv University, Israel

Itay Cohen
Tel Aviv University, Israel

Gal Maor
Tel Aviv University, Israel

Yuval Peled
The Hebrew University of Jerusalem, Israel

Abstract

The notion of the derandomized square of two graphs, denoted as $G ⓢ H$, was introduced by Rozenman and Vadhan as they rederived Reingold's Theorem, **SL** = **L**. This pseudorandom primitive, closely related to the Zig-Zag product, plays a crucial role in recent advancements on space-bounded derandomization. For this and other reasons, understanding the spectral expansion $\lambda(G ⓢ H)$ becomes paramount. Rozenman and Vadhan derived an upper bound for $\lambda(G ⓢ H)$ in terms of the spectral expansions of the individual graphs, $\lambda(G)$ and $\lambda(H)$. They also proved their bound is optimal if the only information incorporated to the bound is the spectral expansion of the two graphs.

The objective of this work is to gain deeper insights into the behavior of derandomized squaring by taking into account the *entire spectrum* of H, where we focus on a vertex-transitive c-regular H. Utilizing deep results from analytic combinatorics, we establish a lower bound on $\lambda(G ⓢ H)$ that applies universally to all graphs G. Our work reveals that the bound is the minimum value of the function

$$d \cdot x - \frac{d(d-1)\chi_x(H)}{\chi'_x(H)}$$

in the domain (c, ∞), where $\chi_x(H)$ is the characteristic polynomial of the d-vertex graph H. This bound lies far below the known upper bound for $\lambda(G ⓢ H)$ for most reasonable choices for H. Empirical evidence suggests that our lower bound is optimal. We support the tightness of our lower bound by showing that the bound is tight for a class of graphs which exhibit local behavior similar to a derandomized squaring operation with H. To this end, we make use of finite free probability theory.

In our second result, we resolve an open question posed by Cohen and Maor (STOC 2023) and establish a lower bound for the spectral expansion of rotating expanders. These graphs are constructed by taking a random walk with vertex permutations occurring after each step. We prove that Cohen and Maor's construction is essentially optimal. Unlike our results on derandomized squaring, the proof in this instance relies solely on combinatorial methods. The key insight lies in establishing a connection between random walks on graph products and the Fuss-Catalan numbers.

2012 ACM Subject Classification Mathematics of computing → Spectra of graphs

Keywords and phrases Derandomized Squaring, Spectral Graph Theory, Analytic Combinatorics

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.40

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2023/183/`

Funding *Gil Cohen*: Supported by ERC starting grant 949499 and by the Israel Science Foundation grant 1569/18.



© Gil Cohen, Itay Cohen, Gal Maor, and Yuval Peled;
licensed under Creative Commons License CC-BY 4.0



Itay Cohen: Supported by ERC starting grant 949499 and by the Israel Science Foundation grant 1569/18.
Gal Maor: Supported by ERC starting grant 949499 and by the Israel Science Foundation grant 1569/18.

1 Introduction

Expander graphs have played a crucial role in essentially all areas within theoretical computer science as well as in coding theory, and cryptography, among others. Their utility stems to a large extent from the ability to interpret expansion from various perspectives, be it combinatorial, probabilistic, or linear algebraic. This multifaceted understanding offers a unique advantage: it enables the otherwise challenging inference of combinatorial attributes of graphs by examining the spectral properties of related operators.

We briefly recall the notion of *spectral expansion*. Let G be an undirected d-regular graph on n vertices with adjacency matrix $\mathbf{A}$. Since G is undirected, $\mathbf{A}$ is symmetric and so its spectrum is real-valued. We denote the eigenvalues of $\mathbf{A}$ by $d = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. The spectral expansion of G, denoted as $\lambda(G)$, is given by $\max(\lambda_2, |\lambda_n|)$. We further denote the normalized spectral expansion of G by $\omega(G) = \frac{\lambda(G)}{d} \in [0, 1]$. We alternate between the two variants–the normalized and the unnormalized–depending on context.

An *expander* is a graph G with a normalized spectral expansion $\omega(G)$ that is bounded away from 1 [1]. However, for a typical application of expander graphs one "pays" a cost that increases with the degree d and has an "error" that vanishes as $\omega(G) \to 0$. This raises the question of what is the lowest possible value of $\omega(G)$ attainable by d-regular graphs. From the Alon-Boppana bound [30], which is usually stated in terms of $\lambda(G)$, it follows that for every $\varepsilon > 0$ there are only finitely many d-regular graphs G with $\lambda(G) \leq 2\sqrt{d-1} - \varepsilon$. A d-regular graph G satisfying $\lambda(G) \leq 2\sqrt{d-1}$ is called a *Ramanujan graph*. Over the past several decades, Ramanujan graphs have been a focal point of research. The constructions of Ramanujan graphs and their variants lean on profound number theoretic results [19, 24, 27] (see also [18]), or is rooted in deep analytical methods and on the accompanied technique of polynomial interlacing [20, 23, 22, 13, 16].

In their highly influential paper [34], Reingold, Vadhan, and Wigderson introduced the Zig-Zag product which enabled them to obtain a combinatorial construction of expander graphs by elementary means. While the expanders that were constructed were not quite close to Ramanujan, the fact that the construction is combinatorial and highly flexible made the Zig-Zag product extremely useful. Indeed, no long after, Reingold [33] based his breakthrough result, $\mathbf{SL} = \mathbf{L}$, on the Zig-Zag product, not for constructing expanders per se but for the purpose of "transforming" a given graph to an expander while maintaining its connected components structure. In a subsequent work, Ben-Aroya and Ta-Shma [5] put forth an improved variant of the Zig-Zag product, dubbed the *wide-replacement product*, that enabled the combinatorial construction of graphs that come quite close to Ramanujan. That variant was key in a recent breakthrough by Ta-Shma who constructed near-optimal small-bias sets [36]. Several other expander construction paradigms have been put forth in the literature, e.g., [6, 26]. We refer the reader to the excellent survey by Hoory, Linial, and Wigderson [17] for a comprehensive exposition on expander graphs.

[1] To be more precise, it is common to consider a family of graphs in this context. However, we will exclude this technical detail from our discussion for simplicity.

1.1 Derandomized squaring

Not long after the work of Reingold [33], Rozenman and Vadhan [35] introduced a close sibling to the Zig-Zag product, dubbed *derandomized squaring*, which we describe soon. This operation can be "cleaner" than the Zig-Zag product in some settings, in particular for their rederivation of Reingold's result, though the two operations are tightly connected. In the past few years, derandomized squaring has gained significant traction in the space-bounded derandomization literature (see, e.g., [28, 1, 2, 11]) mostly since it facilitated the adaptation of ideas from the realm of fast Laplacian solvers into the domain of space-bounded derandomization. For these reasons, in this paper we focus on the operation of derandomized squaring though we are confident that our techniques can be extended to other operations such as the Zig-Zag product and the wide-replacement product.

Squaring a graph is an easy way of improving its normalized spectral expansion. The square of a graph G, denoted as G^2, is the graph on the same vertex set that has an edge between a pair of vertices for every length-2 path in G between the vertices. Clearly, if $\mathbf{A}$ is the adjacency matrix of G then $\mathbf{A}^2$ is the adjacency matrix of G^2. Hence, $\omega(G^2) = \omega(G)^2$. However, if G is d-regular, G^2 is a d^2-regular graph. As a result, the degree growth associated with squaring the graph often surpasses the advantages gained from reducing the normalized spectral expansion. The purpose of derandomized squaring is to obtain a comparable improvement to the normalized spectral expansion without blowing up the degree by a quadratic factor.

From the view point of a vertex v of G, in the graph G^2, the neighbors of v are all connected to each other. That is, G^2 is obtained by adding copies of the complete graph with self-loops, one copy for each vertex v, where the complete graph associated with v is placed on the neighbors of v. Let H be a graph on d vertices, where we focus on the case in which H is vertex-transitive, and denote the degree of a vertex in H by c. The derandomized square of G and H, denoted as $G \circledS H$, is defined by replacing each such copy of the complete graph with a copy of H. Note that $G \circledS H$ is a D-regular graph where $D = dc$. Formally, the derandomized squaring, like the Zig-Zag product, requires working with edge-labeled graphs, but we sidestep this technicality. For the reader that is familiar with these intricacies, we remark that, for simplicity, in this extended abstract we circumvent labeling issues by assuming that G is given as the union of d perfect matchings, though this condition can be relaxed.

Given that an expander H approximates the complete graph, one is correct to expect that the derandomized square $G \circledS H$ approximates G^2 for every graph G. Rozenman and Vadhan formalized this intuition with regards to the normalized spectral expansion by establishing the bound

$$\omega(G \circledS H) \leq (1 - \omega(H))\,\omega(G)^2 + \omega(H) \leq \omega(G)^2 + \omega(H). \tag{1.1}$$

How tight is this bound? This is a somewhat subtle question. Rozenman and Vadhan proved that the bound is tight *as a function of* $\omega(G)$ and $\omega(H)$, however, it is certainly conceivable that a superior bound might be achieved if one incorporates more information about the graphs beyond just their spectral expansions into the bound. In particular, if we fix H and consider the mapping $\lambda(\cdot \circledS H)$ which maps every d-regular graph G to $\lambda(G \circledS H)$, then it is interesting to ask how strong a bound can be obtained on this mapping as a function of the entire spectrum of H.

Given the significance of the derandomized squaring operation, and the related Zig Zag product as well as the wide-replacement product, a substantial improvement to the bound could profoundly impact our understanding on several fundamental problems, including

in space-bounded derandomization and in coding theory. For example, in the context of space-bounded derandomization such a result may reduce the seed length of PRGs or weighted PRGs (see [9, 10, 11] and references therein). Such a result may also lead to a better construction of small-bias sets in which the wide-replacement product is used for bias-reduction. Therefore, gaining a thorough understanding of derandomized squaring is highly motivated from the theoretical computer science standpoint.

1.2 Two case studies

Before presenting our results, which we view as a first step towards the above goal, we wish to highlight the gap between the Rozenman-Vadhan bound, as given in Equation (1.1), and the "true" behavior of the derandomized squaring operation. We do so by considering two case studies, starting with $H = \mathrm{K}_{c,c}$, the complete $c = \frac{d}{2}$-regular bipartite graph on d vertices.

1.2.1 Derandomized squaring with the complete regular bipartite graph

Since $H = \mathrm{K}_{c,c}$ is bipartite, $\omega(H) = 1$, and so the bound given by Equation (1.1) becomes trivial, $\omega(G \circledS H) \leq 1$ for all d-regular graphs G. For this special case, one can get a nontrivial bound by elementary means by incorporating some information on G. To see this, assume that G is the union of two c-regular Ramanujan graphs on n vertices, whose adjacency matrices are denoted $\mathbf{B}$ and $\mathbf{R}$, respectively. That is, the adjacency matrix of G is given by $\mathbf{A} = \mathbf{B} + \mathbf{R}$. Thus, it can be shown that the adjacency matrix of $G \circledS H$ can be expressed as $\mathbf{RB} + \mathbf{BR}$ [2], and so by the Courant-Fischer Theorem,

$$\lambda(G \circledS H) = 2 \cdot \max_{x \perp \mathbf{1}} \frac{x^{\mathsf{T}}\mathbf{BR}x}{x^{\mathsf{T}}x} \leq 2\left(2\sqrt{\frac{d}{2} - 1}\right)^2 \leq \sqrt{32}\sqrt{D-1} \approx 5.66\sqrt{D-1},$$

where $D = \frac{d^2}{2}$ is the regularity of $G \circledS H$.

Although the above bound on $\lambda(G \circledS H)$ certainly beats the trivial bound, D, it still seems to undersell the typical behavior of $G \circledS H$. In fact, by sampling [3] a random d-regular graph G and evaluating $\lambda(G \circledS H)$, one can verify that for a sufficiently large d, the value of $\lambda(G \circledS H)$ distributes around $2.35\sqrt{D-1}$. But where does the 2.35 value originate? How can this number be determined based on our selected H? Jumping the gun, the analytical tool we introduce *predicts* that the exact value, for this H, is

$$\frac{1}{2}\sqrt{11 + 5\sqrt{5}} \approx 2.35. \tag{1.2}$$

By saying that we "predict" this bound captures the true behavior of derandomized squaring with $H = \mathrm{K}_{c,c}$, we mean the following: First, we prove the aforementioned value to be a lower bound on $\lambda(G \circledS H)$ for *every* d-regular graph G; Second, we prove the existence of infinitely many graphs that meet this bound. These graphs exhibit a local structure resembling a derandomized square with H. Lastly, our predictions align with every experiment we made for every graph H and when G is sampled uniformly at random. We provide further details in Section 2, where our results are formally presented.

---

[2] With regards to the edge labeling, for the last statement to hold we assume that neighbors $1, \ldots, \frac{d}{2}$ of every vertex are those coming from $\mathbf{B}$ and the remaining neighbors $\frac{d}{2} + 1, \ldots, d$ are coming from $\mathbf{R}$.

[3] Our sampling is done by taking the union of d uniformly random and independent perfect matchings, where edges that are sampled multiple times are counted with the respective multiplicity.

1.2.2 Derandomized squaring with a Paley graph

In the aforementioned example, the Rozenman-Vadhan bound was non-informative. It is worth noting that it is not just in these instances where the typical performance of the derandomized square surpasses the predictions of the bound. To give one such example, consider a Paley graph on d vertices, denoted as Pal_d. For a typical d-regular graph G, the limit behavior as $d \to \infty$ of the spectral expansion $\lambda(G ⓢ \mathrm{Pal}_d)$, when properly normalized, is predicted by our analytic tool to equal

$$\lim_{d\to\infty} \frac{\lambda(G ⓢ \mathrm{Pal}_d)}{\sqrt{D-1}} = \frac{1+\sqrt{13+16\sqrt{2}}}{\sqrt{8}} \approx 2.46, \tag{1.3}$$

where $D = \frac{d(d-1)}{2}$ is the degree $G ⓢ \mathrm{Pal}_d$. This should be compared with a bound of 2, the least possible value given the Alon-Boppana bound. Additionally, it should be compared with Equation (1.1), which, irrespective of the choice of G, cannot produce a bound lower than $O(D^{1/4})$.

From the preceding discussion and, in particular, the two case studies, a key question lingers: Is there an exact formula or efficient method that enables us to compute, and more importantly, to gain insight on the spectral expansion of the operation of derandomized squaring with H?

2 Our Results

As our case studies suggest, the spectral expansion of the graphs G and H might not adequately represent the spectral expansion of their derandomized square. In this paper we initiate the study of the following question:

▶ **Main Question.** *What is the "true" behavior of the spectral expansion of derandomized squaring?*

We turn to give a brief summary of our results. We elaborate further on each of these results in the subsequent sections, Sections 2.1–2.3.

Limitations of derandomized squaring

Our first result is a lower bound on $\lambda(G ⓢ H)$, factoring in the full spectrum of H, which holds for *every* graph G. Encoding this spectrum by the characteristic polynomial of H, denoted as $\chi_x(H)$, our work reveals that the bound is the minimum value of the function

$$d \cdot x - \frac{d(d-1)\chi_x(H)}{\chi'_x(H)} \tag{2.1}$$

in the domain (c, ∞), where recall that d and c are the degrees of G and H, respectively. Our proof leans on deep results from analytic combinatorics and the symbolic method. Although this bound may look more complicated than Equation (1.1), it is easy to calculate for various explicit choices of H, and easy to bound for families of graphs, as we will examplify in Section 2.1 (a more comprehensive review of examples, including the ones discussed in Section 1.2, is done in Section 4).

Based on our empirical experiments, it appears that our lower bound is tight in a strong sense, namely, for *every* vertex-transitive graph H and for a *typical* graph G. However, a definitive proof of the bound's tightness eludes us in general. In spite of this, we have

made notable progress by first proving tightness of the bound for particular choices of H, and second by proving its tightness for a broader class of graphs, we term H-local graphs. For obtaining this result we make use of finite free probability theory and the accompanied technique of interlacing. These were instrumental in the seminal works of Marcus, Spielman, and Srivastava [21, 22, 23] who introduced these techniques for their study of bipartite Ramanujan graphs. We elaborate on this in Section 2.2.

Comparison to known bounds

The lower bound described above requires solving a separate polynomial equation for every choice of H. However, restricting the discussion to *reasonable* choices of H–for example, graphs H having no self loops–we show universally that the lower bound for $\omega(G \circledS H)$ lies within the range $\left(\frac{2}{\sqrt{dc}}, \frac{3}{\sqrt{dc}}\right)$, very close to the Alon-Boppana bound and significantly distant from Equation (1.1) that suggests at best that $\omega(G \circledS H) = O\left(\frac{1}{\sqrt{c}}\right)$. As discussed above, we also expect our lower bound to represent the behavior of $G \circledS H$ for a typical choice of G. Note that the restriction of H to simple graphs makes sense, as in all applications of derandomized squaring we are aware of, H is for our choosing.

The proof of Rozenman and Vadhan [35] for the tightness of Equation (1.1) involves a choice of H which is a graph with many self loops. Moreover, our lower bound applied to this specific choice (see Section 4.1.3) exactly yields the formula given by Equation (1.1). This suggests that a better upper bound should hold for particular choices of H (potentially under some assumptions on G). We leave these intriguing open questions for future work.

A lower bound for rotating expanders

In the process of establishing our lower bound on the spectral expansion of derandomized squaring, we address an open problem concerning the spectral expansion of *rotating expanders* [12]. In this recent paper, random walks on expanders were studied, wherein a permutation is applied to the vertices following each step. The objective of this approach is to mitigate the inherent exponential deterioration of the spectral expansion with respect to the length of the walk. Indeed, the authors proved that by using a carefully chosen permutation sequence, the deterioration can be reduced from exponential to linear. The authors left open the question of whether their construction is optimal.

In this work, we resolve this question by proving the optimality of their construction. More generally, we prove that a graph which is constructed as a graph product is inherently far from Ramanujan. Our key observation lies in relating the problem with the Fuss-Catalan numbers which generalize the Catalan numbers that emerge when bounding the spectral expansion of d-regular graphs. We elaborate on this in Section 2.3, where we also give the necessary background on rotating expanders.

A broader perspective: beyond spectral expansion

Almost all of the numerous works and applications of spectral expanders in theoretical computer science, indeed, the very definition of a spectral expander G, rely on the notion of the spectral expansion, $\lambda(G)$. Only a few instances utilize the entire spectrum of G, which holds significantly more, and sometimes vital, information about the graph. In their seminal series of works, Marcus, Spielman, and Srivastava developed finite free probability as a framework to handle the full spectrum of a graph. As mentioned, this enabled them to establish the existence of bipartite Ramanujan graphs of all sizes and degrees.

Our current work serves as a further exploration into analyzing graphs beyond their spectral expansion. Instead of aiming to construct expanders, our objective is to achieve a deeper insight into the derandomized squaring operation. While we primarily target the spectral expansion of the derandomized square $G \circledS H$, our approach involves leveraging the entire spectrum of H for establishing our bounds. In addition to our use of finite free probability, we employ deep results from analytic combinatorics, and the framework offered by the symbolic method. We posit that working with the full spectrum of a graph could yield significant results and improvements for various problems in theoretical computer science, and we believe that the deep results we use could be advantageous in other scenarios where analyzing the full spectrum is desired.

Related work

Already at this point, prior to delving into the formal details concerning our results, we would like to highlight some related work. Our study of the graph $G \circledS H$ for a d-regular graph G is done by considering the graph $\mathcal{T}_d \circledS H$, where $\mathcal{T}_d$ is the d-ary infinite tree. The study of graphs, which represent the quotient of a given (typically infinite) graph X, has a long history (see [15] as an example). Of particular interest are the extreme graphs, known as *X-Ramanujan graphs*. The reader is referred to [25, 31] and references therein for more details. Our result supporting the tightness of our lower bound, presented in Section 3.2.2, can also be derived from the work of Mohanty and O'Donnell [25]. For completeness and accessibility of our paper, we present the full proof here, as it sheds more light on Equation (2.1) and connections between analytic combinatorics and free probability which we discuss further in Section 2.2.

Regarding our lower bound, the spectral radius of operators associated with infinite graphs has been extensively explored. This is particularly true when these graphs exhibit a well-defined group-theoretic structure. Analytic combinatorics has a well-established presence in this context [37]. Finally, it is noteworthy that the Fuss-Catalan numbers are significant in free probability theory and have known associations with the product of certain random matrices [32].

2.1 Limitations of derandomized squaring

Let H be a vertex-transitive graph on d vertices. Recall that, throughout, c denotes the degree of a vertex in H. In this section we state our result regarding the lower bound on $\lambda(G \circledS H)$ which holds for every d-regular graph G. As previously suggested, our approach integrates the complete spectrum of H into the bound. This integration is accomplished by encoding the spectrum through the characteristic polynomial of H-s adjacency matrix, denoted as $\chi_x(H) = \prod_{i=1}^{d}(x - \lambda_i)$, where, as before, $c = \lambda_1 \geq \cdots \geq \lambda_d$ are the corresponding eigenvalues.

▶ **Theorem 1.** *Let H be a vertex-transitive c-regular graph on $d \geq 3$ vertices, where $c \geq 1$. Let x_0 be the largest real solution to the polynomial equation*

$$(d-1)\chi_x(H)\chi''_x(H) = (d-2)\chi'_x(H)^2. \tag{2.2}$$

Then, for every d-regular graph G on n vertices, $\lambda(G \circledS H) \geq \Lambda_H - o_n(1)$, where

$$\Lambda_H \triangleq d\left(x_0 - (d-1)\frac{\chi_{x_0}(H)}{\chi'_{x_0}(H)}\right). \tag{2.3}$$

As shown in the proof of Theorem 1, despite the polynomial equation from Equation (2.2) usually yielding complex solutions, there is always at least one real solution. Experiments overwhelmingly suggest that the bound accurately reflects the behavior of $\lambda(G \circledS H)$ for a typical graph G. Having this in mind, we may posit that the structure of Equations (2.2) and (2.3), namely, $(d-1)\Phi\Phi'' = (d-2)(\Phi')^2$, and the accompanied expression $d\left(x - (d-1)\frac{\Phi(x)}{\Phi'(x)}\right)$, epitomize the derandomized squaring operation with a typical d-regular graph. By setting $\Phi = \chi(H)$, we incorporate details about H.

Although the proof of Theorem 1 leans on deep results from analytic combinatorics and the symbolic method, employing the theorem remains elementary. However, as perhaps anticipated, it is not as direct as the Rozenman-Vadhan bound from Equation (1.1), but rather it requires finding the largest real solution to a polynomial equation.

Before proceeding further, we introduce the following notation. Recall that $G \circledS H$ is D-regular where $D = cd$. We define $\kappa_H = \frac{\Lambda_H}{\sqrt{D-1}}$, and note that, due to the Alon-Boppana bound, $\kappa_H \in \left[2, \frac{D}{\sqrt{D-1}}\right] \approx \left[2, \sqrt{D}\right]$.

Equivalent reformulations of Theorem 1

One can alternatively recast our procedure for finding a lower bound for $\lambda(G \circledS H)$, as given by Theorem 1, in several equivalent ways, as we describe next.

The Cauchy transform is a useful analytic tool which we will make an extensive use of in this paper. For a graph H on d vertices, the Cauchy transform takes a simple form and is given by

$$\mathcal{G}_H(x) = \frac{1}{d} \cdot \frac{\chi'_x(H)}{\chi_x(H)} = \frac{1}{d} \sum_{i=1}^{d} \frac{1}{x - \lambda_i}. \tag{2.4}$$

Using the Cauchy transform we can reformulate Theorem 1 as follows.

▶ **Theorem 2** (Recasting Theorem 1 in terms of the Cauchy transform). *Let H be a vertex-transitive c-regular graph on $d \geq 3$ vertices, where $c \geq 1$. Let x_0 be the unique positive real solution to the equation*

$$\frac{d}{d-1}\mathcal{G}_H(x)^2 + \mathcal{G}'_H(x) = 0. \tag{2.5}$$

Then, for every d-regular graph G on n vertices, $\lambda(G \circledS H) \geq \Lambda_H - o_n(1)$, where

$$\Lambda_H = d \cdot x_0 - \frac{d-1}{\mathcal{G}_H(x_0)}. \tag{2.6}$$

We emphasize that, as demonstrated in the proof of Theorem 2, there always exists a positive real solution to Equation (2.5) and it is unique.

By defining $\psi_H(x) = d \cdot x - \frac{d-1}{\mathcal{G}_H(x)}$, it can further be shown that

$$\Lambda_H = \min_{x > c} \psi_H(x), \tag{2.7}$$

where the minimal point $x_0 > c$ exists and is unique (see the full version of the paper for the details).

2.2 Tightness of the lower bound

As previously discussed, we have yet to establish that the lower bound provided by Theorem 1 is tight in general. However, empirical results strongly suggest its accuracy. Specifically, we believe this to be true for *every* vertex-transitive graph H on d vertices and for a *typical* d-regular graph G. In light of the evidence supporting this assertion, as we discuss next, we will formulate it as a conjecture in Section 2.4. In order to state our analytic evidence for the tightness of our bound, we observe that from the perspective of a vertex v in $G \circledS H$, the vertex v participates in d instances of H. This is because each of the d neighbors of v positions it within a copy of H. We call a graph that has the above local property an *H-local graph* (see the full version of this paper for the formal definition). The following theorem formalizes the evidence we have gathered for the tightness of our lower bound.

▶ **Theorem 3.** *For every vertex-transitive graph H on $d \geq 3$ vertices and for every $n \geq 1$, there exists an H-local graph X_H on nd vertices such that $\lambda_2(X_H) \leq \Lambda_H$.*

In addition to Theorem 3, we prove the optimality of our lower bound for $\lambda(G \circledS H)$ in three specific cases of H: the clique with self-loops, where $G \circledS H$ corresponds to the actual squaring operation; the clique without self-loops, which corresponds to a non-backtracking length-2 random walk; and lastly, the graph employed in the Rozenman-Vadhan bound's tightness result. We provide further details on these cases in Section 4.1. Going back to Theorem 3, note that we manage to bound only the second-largest eigenvalue, $\lambda_2(X)$, rather than the spectral expansion $\lambda(X) = \max(\lambda_2(X), |\lambda_n(X)|)$. Graphs with such property are termed *one-sided spectral expanders*. These graphs are suitable for numerous applications, primarily due to the fact that this property alone suffices for the Alon-Chung Lemma [3].

The proof of Theorem 3 leverages finite free probability and the interlacing technique that were developed by Marcus, Spielman, and Srivastava [21, 22, 23]. We provide a high-level overview for the proof of Theorem 3 in Section 3.2, however, already here we emphasize that the fact that our lower bound, which is based on results from analytic combinatorics, matches our upper bound which is rooted in tools from free probably theory is an instantiation of a deep connection between the two fields. This has to do with the fact that one combinatorial proof for the Lagrange inversion formula–a tool used under the hood in our lower bound– makes use of Lukasiewicz paths that in turn are tightly connected to the lattice of non-crossing partitions which is at the heart of free probability theory. The reader is referred to Chapter 16 in the excellent book by Nica and Speicher [29] to learn more about this connection, though for our purpose, of studying the derandomized squaring operation, we give a direct and self-contained proof in the full version of the paper.

2.3 Lower bound on the spectral expansion of rotating expanders

A "standard" length-t random walk on a graph G is analyzed by considering the power graph, denoted as G^t, generalizing the square of the graph G^2 which was discussed so far. This graph encodes the number of length-t walks by introducing an edge for each such walk between the two corresponding vertices. It is easy to see that if $\mathbf{A}$ is the adjacency matrix of G, then the matrix $\mathbf{A}^t$ is the adjacency matrix of G^t. Consequently, the spectral expansion of G^t, which is the most pertinent quantity when examining length-t random walks on G, is given by $\lambda(G^t) = \lambda(G)^t$. In particular, if G is a d-regular Ramanujan graph, then $\lambda(G^t) = 2^{\Omega(t)}\sqrt{D-1}$, where $D = d^t$ is the degree of G^t. Therefore, even if G is initially Ramanujan, the power graph is exponentially distant, in t, from Ramanujan.

With an eye towards potential applications to theoretical computer science, Cohen and Maor [12] proposed that permuting the vertices after each step (in a palindrome fashion to result in an undirected graph) can circumvent this exponential deterioration. More

precisely, the authors proved that for every d-regular Ramanujan graph G with adjacency matrix $\mathbf{A}$, and for every integer $t \geq 2$, there exists a sequence of permutation matrices $\mathbf{P} = (\mathbf{P}_1, \ldots, \mathbf{P}_{t-1})$ such that the graph $G_{\mathbf{P}}$, whose adjacency matrix is given by

$$\mathbf{A}_{\mathbf{P}} = \mathbf{A}\mathbf{P}_{t-1} \cdots \mathbf{A}\mathbf{P}_1\mathbf{A}^2\mathbf{P}_1^{\mathsf{T}}\mathbf{A} \cdots \mathbf{P}_{t-1}^{\mathsf{T}}\mathbf{A},$$

has spectral expansion

$$\lambda(G_{\mathbf{P}}) \leq \left(1 + \frac{1}{t}\right)^t (t+1)\sqrt{D} + o(1), \tag{2.8}$$

where $D = d^{2t}$ is the degree of $G_{\mathbf{P}}$ and the $o(1)$ term is a quantity that vanishes exponentially fast with the girth of G, and should be ignored in this introductory section. Specifically, by permuting the vertices after each step using suitable permutations, the deterioration is reduced from exponential to linear in t.

An open problem left in [12] is to establish a lower bound on the spectral expansion of $G_{\mathbf{P}}$ that is applicable for any permutation sequence $\mathbf{P}$. Specifically, the authors left open the question of whether the linear dependence in t is optimal. Experimental results suggest that for a typical $\mathbf{P}$, Equation (2.8) holds with equality, up to the vanishing $o(1)$ term. However, it is entirely plausible that the typical behavior does not accurately represent the behavior of the optimal permutation sequence $\mathbf{P}$. The logic would be that for a graph with substantial structure, such as a Cayley graph, a permutation sequence that takes into account the structure of the underlying group and the set of generators may yield a superior spectral expansion. However, in this work we resolve this open problem by proving that the bound is indeed tight.

▶ **Theorem 4.** *For every d-regular graph G and for every permutation sequence $\mathbf{P} = (\mathbf{P}_1, \ldots, \mathbf{P}_{t-1})$,*

$$\lambda(G_{\mathbf{P}}) \geq \left(1 + \frac{1}{t}\right)^t (t+1)\sqrt{D} - o(1).$$

In fact, our lower bound applies to the product of *any* d-regular graphs, not only isomorphic graphs as used in the construction of $G_{\mathbf{P}}$. For $t = 1$, where no actual product is involved, this essentially aligns with the Alon-Boppana bound. As suggested by Theorem 4 (ignoring the $o(1)$ term), we have $\lambda(G_{\mathbf{P}}) = \lambda(G^2) \geq 4\sqrt{D}$. However, for $t = 2$, the bound increases to $\lambda(G_{\mathbf{P}}) \geq \frac{3^3}{2^2}\sqrt{D} = 6.75\sqrt{D}$, and for $t = 3$, it further deteriorates to $\lambda(G_{\mathbf{P}}) \geq \frac{4^4}{3^3}\sqrt{D} \approx 9.48\sqrt{D}$. As indicated, the gap for graph products increases linearly with the number of graphs involved, making them inherently far from Ramanujan.

2.4 Two conjectures and open problems

Given our results and the above discussion, we wish to put forth two conjectures that capture different aspects of the tightness of our lower bound as given by Theorem 1. These are analog to fundamental questions on Ramanujan graphs where Theorem 1 plays in this analogy the role of the Alon-Boppana bound.

▶ **Conjecture 5.** *For every vertex-transitive graph H, $\lambda(G \circledS H) \leq \Lambda_H$ holds for infinitely many graphs G.*

Conjecture 5 is analog to the fundamental question regarding the existence of Ramanujan graphs which has received significant attention in the literature. Resolving Conjecture 5 with respect to $\lambda_2(G \circledS H)$ would be interesting as well. Our second conjecture focuses on the

typical behavior, and is analogous to Friedman's resolution [14] of Alon's conjecture [4] (see also [7]). We first introduce the following notation: For an even integer n and for an integer d, we let $\mathcal{M}_{n,d}$ denote the distribution over d-regular graphs on n vertices that are sampled by taking the union of d uniformly random and independent perfect matchings, where edges that are sampled multiple times are counted with the respective multiplicity.

▶ **Conjecture 6.** *For every vertex-transitive graph H on d vertices and for every $\varepsilon > 0$,*

$$\Pr_{G\sim\mathcal{M}_{n,d}}[\lambda(G\circledS H) \geq \Lambda_H + \varepsilon] = o_n(1).$$

Note that a similar statement is proven in [8] for a wide variety of graph distributions. However, these do not include the one suggested above for graphs of the form $G\circledS H$.

In addition to the conjectures previously discussed, our research raises several intriguing questions. An obvious open problem is the generalization of our results to non-vertex-transitive graphs. For potential theoretical computer science applications, it would be pertinent to identify conditions that a pair of graphs G and H satisfy so that the spectral expansion $\lambda(G\circledS H)$ is close to our lower bound or, at a minimum, substantially improves upon the Rozenman-Vadhan bound. Once this aspect is clearer, problems regarding the explicitness can be addressed. To give just one additional research question, we believe that the extension of our techniques to additional graph operations, including the Zig-Zag product and the wide-replacement product, is feasible. We defer this exploration to future research.

Organization of the rest of the paper

In Section 3 we give a high-level proof overview of our results on the derandomized squaring operator. The detailed proofs of the theorems stated above are provided in the full version of this paper. In Section 4 we apply our results to interesting graph families, and prove our universal bound on κ_H for simple graphs.

3 Proof Overview

In this section, we provide an informal overview of the proofs for our results. We begin with our lower bound for the spectral expansion of derandomized squaring, as given by Theorem 1 (outlined in Section 3.1). Our proof relies on the symbolic method and leverages results from analytic combinatorics, both of which we introduce and explain in the full version of this paper. Additionally, we briefly outline the proof for our evidence regarding the tightness of our lower bound, as stated in Theorem 3, in Section 3.2. In that section, we provide the necessary background on finite free probability, which is essential for understanding the proof.

3.1 Limitations of derandomized squaring

As before, let G be a d-regular graph and H a vertex-transitive c-regular graph on d vertices. In this section we sketch the proof for our lower bound on $\lambda(G\circledS H)$, as stated in Theorem 1. Our starting point is standard, relying on the trace method which asserts that $\lambda(G\circledS H)$ is lower bounded by roughly $c_\ell(G\circledS H)^{1/\ell}$ for every $\ell > 0$, where $c_\ell(G\circledS H)$ is the number of length-ℓ cycles that originate at some fixed vertex v of $G\circledS H$. Thus, the task at hand is to compute, or at least lower bound $c_\ell(G\circledS H)$, where we will choose ℓ to be sufficiently large. A common strategy for this is to consider a suitable infinite cover of the graph of interest,

$G \circledS H$ in our case, which we take to be $\mathcal{T}_d \circledS H$, where $\mathcal{T}_d$ is the d-regular infinite tree. Indeed, for every ℓ, every length-ℓ cycle in $\mathcal{T}_d \circledS H$ that originated at some fixed vertex induces a unique cycle in $G \circledS H$, initiated at some fixed vertex, and so $c_\ell(G \circledS H) \geq c_\ell(\mathcal{T}_d \circledS H)$.

We obtain an accurate estimate on $c_\ell(\mathcal{T}_d \circledS H)$ by first expressing the combinatorial class of cycles in $\mathcal{T}_d \circledS H$ using the symbolic method, from which we immediately derive a functional equation that is satisfied by the corresponding generating function. We then use results from analytic combinatorics to get the desired estimate on the coefficients of the latter. The symbolic method, a prominent combinatorial theory, allows one to deduce a functional equation that is satisfied by the class's generating function straight from its specification. Following this, in Section 3.1.1, we utilize the symbolic method to define the cycle class in $\mathcal{T}_d \circledS H$ and from there, derive a functional equation that is satisfied by the associated generating function.

With the functional equation in hand, our objective is to deduce estimates of its coefficients. To achieve the estimate, we employ deep results from analytic combinatorics. These treat the functional equation as a meromorphic function, considering its singularities to determine the bound.

3.1.1 The functional equation for derandomized squaring

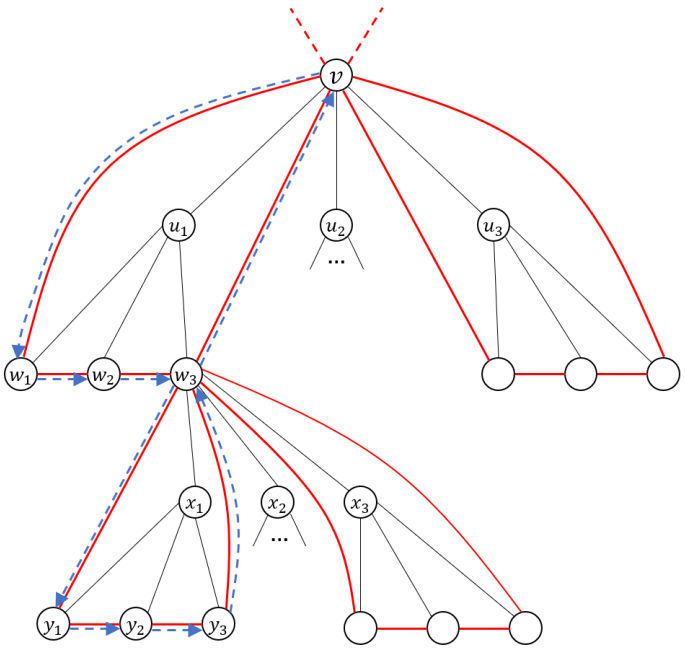


Figure 1 Cycles in $\mathcal{T}_4 \circledS C_4$. The black edges represent the edges of $\mathcal{T}_4$, while the red edges represent those of $\mathcal{T}_4 \circledS C_4$. Dashed red edges indicate the truncated edges. Edges that are irrelevant to the cycles of v (e.g., (u_1, x_1)) have not been included in the figure. The blue cycle within $\mathcal{T}_4 \circledS C_4$ corresponds to the cycle $(v \to w_1 \to w_2 \to w_3 \to v)$ within the copy of H centered around vertex u_1. In this cycle, the first, second, and last steps are substituted with the pair $(\to, \epsilon)$, while the third step is substituted with the pair $\big(\to, c\big)$, where c represents a nested cycle from $\mathcal{C}_{\mathcal{T}_d \circledS H}$ (specifically, $c = (w_3 \to y_1 \to y_2 \to y_3 \to w_3)$).

Let H be a vertex-transitive graph on d vertices. Define $\mathcal{C}_{\mathcal{T}_d \circledS H}$ as the combinatorial class of cycles in $\mathcal{T}_d \circledS H$ that originate at the root. As previously mentioned, the size function corresponds to the cycle's length. To prevent double-counting, we exclude the empty cycle

from this class. When expressing the class using the symbolic method, we use $\mathcal{S}_H$ to represent the combinatorial class of nonempty cycles in H that only revisit the originating vertex upon completing the cycle. Consequently, as detailed below, after truncating one branch of the root, and overriding the class definition of $\mathcal{C}_{\mathcal{T}_d \circledS H}$ to capture cycles on the truncated graph, it satisfies the recursive relation

$$\mathcal{C}_{\mathcal{T}_d \circledS H} = \{1, \ldots, d-1\} \times \left(\mathcal{S}_H \circ \left(\to \times \left(\mathcal{C}_{\mathcal{T}_d \circledS H} + \epsilon\right)\right)\right). \tag{3.1}$$

Here, $\to$ symbolizes an atom, which we interpret as a step within a cycle in H.

To see this, remember that each child of the root positions the root within a copy of H. Therefore, when describing a non-empty cycle originating at the root, we first select one of its $d-1$ children, which determines the copy of H in which the root is involved. Now, consider any cycle within that copy of H starting at the root, $v_1 \to v_2 \to \cdots \to v_\ell = v_1$. This cycle corresponds to a cycle in $\mathcal{T}_d \circledS H$ that begins at the root in the following manner: After each step $v_i \to v_{i+1}$ on the cycle, we examine the copy of $\mathcal{T}_d \circledS H$ rooted at v_{i+1} and attach a cycle from that copy of $\mathcal{T}_d \circledS H$. When we return to v_{i+1}, we proceed with $v_{i+1} \to v_{i+2}$. Note that attaching an empty cycle is permissible, even though it is not included in $\mathcal{C}_{\mathcal{T}_d \circledS H}$, leading to the addition of the neutral element ϵ in Equation (3.1). Furthermore, after the final step $v_{\ell-1} \to v_\ell = v_1$, we attach another (potentially empty) copy of $\mathcal{C}_{\mathcal{T}_d \circledS H}$ to account for cycles that visit the root more than twice. This rationale underpins our definition of $\mathcal{S}_H$, which is designed to prevent over-counting that would have otherwise occur.

Equation (3.1) directly implies that the generating function, $C_{\mathcal{T}_d \circledS H}(z)$, associated with the class $\mathcal{C}_{\mathcal{T}_d \circledS H}$ satisfies

$$C_{\mathcal{T}_d \circledS H}(z) = (d-1) S_H\left(z\left(C_{\mathcal{T}_d \circledS H}(z) + 1\right)\right), \tag{3.2}$$

where $S_H(z)$ is the generating function corresponding to the class $\mathcal{S}_H$. On its own, this result does not provide much insight, as the functional equation $u = (d-1)S_H(z(u+1))$ tends to be intricate, hindering our ability to extract the coefficients of $C_{\mathcal{T}_d \circledS H}(z)$. For instance, even in the simple case of a length-4 cycle, $H = C_4$, in which case $S_{C_4}(z) = \frac{2z^2}{1-2z^2}$, Equation (3.2) takes the form

$$2z^2 c(z)^3 + 10z^2 c(z)^2 + (14z^2 - 1)c(z) + 6z^2 = 0,$$

where the term $c(z)$ is a shorthand for $C_{\mathcal{T}_d \circledS C_4}(z)$, resulting in a complicated expression for $c(z)$.

To address this challenge, we leverage a deep result from analytic combinatorics. We provide the essential background in the full version of the paper. Subsequently, in Section 3.1.3, we will outline our approach to estimating the coefficients of $C_{\mathcal{T}_d \circledS H}(z)$ using Equation (3.2) as our starting point.

3.1.2 A brief introduction to analytic combinatorics

The symbolic method classifies combinatorial classes into *schemes* based on their shared structures. This approach aims to consolidate solutions to these problems and highlight their interrelations. A notable schema within this framework is termed *smooth inverse-function schema*. These are classes whose generating function $\zeta(z)$ satisfies the functional equation $u = z \cdot \phi(u)$, namely, $\zeta(z) = z \cdot \phi(\zeta(z))$, for some "well-behaved" function $\phi(u)$. By manipulating Equation (3.2), we see that $\mathcal{C}_{\mathcal{T}_d \circledS H}$ is tightly connected to this schema. Indeed, letting $\zeta_H(z) = z\big(C_{\mathcal{T}_d \circledS H}(z) + 1\big)$, we have that $\zeta_H(z) = z \cdot \phi_H(\zeta_H(z))$, where

$$\phi_H(u) = 1 + (d-1)S_H(u). \tag{3.3}$$

Analytic combinatorics provides a method to estimate the coefficients of the generating function for smooth inverse-function schema. This approach is applicable under certain technical conditions on $\phi(u)$, which we hide under the rug in this informal proof overview. The key requirement though is that there is a real positive solution to the *characteristic equation* $\phi(u) = u \cdot \phi'(u)$ within ϕ-s analytic domain around the origin. To denote the k'th coefficient of a function we use the coefficient extractor operator, $[z^k]$. For an analytic function $A(z) = \sum A_k z^k$, we have

$$[z^k]A(z) \triangleq A_k$$

With this, we have the following theorem which is informally stated here (see the full version of the paper for the formal statement).

▶ **Theorem 7.** *Let $\zeta(z)$ belong to the smooth inverse-function schema. Then, with τ the positive root of the corresponding characteristic equation $\phi(u) = u \cdot \phi'(u)$, one has*

$$\left([z^k]\,\zeta(z)\right)^{\frac{1}{k}} \approx \phi'(\tau). \tag{3.4}$$

3.1.3 Proof sketch of Theorem 1

With Theorem 7 in hand, and by the discussion above that led us to Equation (3.3), we are ready to sketch the proof of Theorem 1. In fact, it will be more convenient to consider the variant using the Cauchy transform of H. Since $\mathcal{C}_H = \mathsf{SEQ}(\mathcal{S}_H)$, we have that $S_H(z) = 1 - \frac{1}{C_H(z)}$. It can also be shown that $C_H(z) = \frac{1}{z}\mathcal{G}_H(\frac{1}{z})$ (see the full version of the paper for the proof), and so Equation (3.3) takes on the form

$$\phi_H(z) = d - \frac{(d-1)z}{\mathcal{G}_H\left(\frac{1}{z}\right)}.$$

Through some algebraic manipulations, it becomes evident that the characteristic equation, $\phi_H(z) = z \cdot \phi'_H(z)$, transforms into the form presented in Equation (2.5) from Theorem 2 when z is substituted with its reciprocal. Specifically, the equation becomes

$$\frac{d}{d-1}\mathcal{G}_H\left(\frac{1}{z}\right)^2 + \mathcal{G}'_H\left(\frac{1}{z}\right) = 0.$$

Upon further analytical exploration, it can be shown that this equation has a positive real root $z_0 > d$. Consequently, there exists a positive $\tau < \frac{1}{d} \leq R$, where $\tau = \frac{1}{z_0}$, that satisfies the characteristic equation. This allows us to apply Theorem 7 and derive the sought-after estimate

$$\left[z^k\right] C_H(z)^{1/k} \approx \phi'_H(\tau) = \frac{\phi_H(\tau)}{\tau} = dz_0 - \frac{d-1}{\mathcal{G}_H\left(z_0\right)}.$$

3.2 Matching the lower bound with H-local graphs

As briefly discussed in Section 2.2, the proof of Theorem 3 makes use of finite free probability. Thus, to start with, in Section 3.2.1 we give a brief account of this elegant theory. Then, in Section 3.2.2 we sketch the proof of Theorem 3.

3.2.1 Finite free probability

Free probability is a branch of mathematics, initiated by Voiculescu, that extends classical probability theory into the non-commutative setting. In classical probability, random variables are analyzed using their joint distribution, which encodes the correlations or lack of between them. In contrast, free probability introduces the abstract notion of "freeness" to represent the absence of correlations, appropriately defined, among non-commutative random variables. Free probability theory provides, in particular, tools to analyze the spectrum of the sum and product of two operators, given that these operators are free, using knowledge of their individual spectra. Freeness is an infinite-dimensional phenomena in the sense that a pair of finite-dimensional operators can only be free from one another if one of them is constant. As a result, operators associated with finite graphs cannot be studied directly by free probability theory.

In response to this limitation, Marcus, Spielman, and Srivastava, in their groundbreaking series of works [21, 22, 23], introduced the theory of *finite free probability* along with the associated technique of interlacing. This enabled them to extend some results of free probability to the finite-dimensional setting, especially regarding the spectra of matrix sums and products. While finite free probability may not capture all details of the latter spectra, it provides a one-sided bound on its support. Finite free probability does not depend on the abstract concept of freeness or any analogous notion. Rather, it demonstrates that conjugating a finite operator, $\mathbf{A}$, with an orthogonal matrix – sampled according to the Haar measure on this group – effectively "frees" $\mathbf{A}$ from other operators. We consider a specific application of this principle in the context of operator addition.

▶ **Definition 8** (Definition 2.4 in [22]). Let $\mathbf{A}, \mathbf{B}$ be $d \times d$ real symmetric matrices, with characteristic polynomials $a(x)$ and $b(x)$, respectively. The *additive convolution* of $a(x)$ and $b(x)$ is defined as

$$a(x) \boxplus_d b(x) = \mathop{\mathbf{E}}_{\mathbf{Q}} \chi_x \left(\mathbf{A} + \mathbf{Q}\mathbf{B}\mathbf{Q}^{\mathsf{T}}\right), \tag{3.5}$$

where the expectation is taken over random orthogonal matrices $\mathbf{Q}$ sampled according to the Haar measure on the group of n-dimensional orthogonal matrices.

We note that, while it may not be immediately apparent, the right-hand side of Equation (3.5) depends only on the spectra of $\mathbf{A}$ and $\mathbf{B}$. Consequently, the additive convolution is well-defined. It was also proved in [23] that $a(x) \boxplus_d b(x)$ is real-rooted itself, hence a discussion of bounding its roots is sensible. When d is clear from context, we omit the subscript d in $\boxplus_d$.

The analytic machinery that will allow us to study the additive convolution is the Cauchy transform and its max-inverse (see the full version for details). More precisely, the Cauchy transform of a real-rooted degree d polynomial $p(t) \in \mathbb{R}[t]$, whose roots are $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$, is defined as $\mathcal{G}_p(x) = \frac{1}{d}\sum_{i=1}^{d} \frac{1}{x-\lambda_i}$. Clearly, for an undirected graph H, the Cauchy transform $\mathcal{G}_H(x)$ as defined in Equation (2.4) can be expressed as $\mathcal{G}_p(x)$ where $p(t) = \chi_t(H)$ is the characteristic polynomial of H.

Note that when the Cauchy transform of a polynomial $p(t)$ is restricted to the domain to the right of its rightmost pole, (λ_1, ∞), its range is $(0, \infty)$. Additionally, $\mathcal{G}_p(x)$ is monotonically decreasing within this domain. With this in mind, one can define $\mathcal{K}_p : (0, \infty) \to (\lambda_1, \infty)$ to be the inverse of $\mathcal{G}_p(x)$ when restricted to the latter domain. In other words, $\mathcal{K}_p$ is the max-inverse of $\mathcal{G}_p$. Particularly, for every $y \in (0, \infty)$, $\mathcal{K}_p(y)$ provides an upper bound on the largest root, λ_1, of $p(t)$. The key feature of the $\mathcal{K}$-transform is that it behaves very-well under additive convolution.

▶ **Theorem 9** (Theorem 1.12 in [23]). *For all $d \times d$ real symmetric matrices $\mathbf{A}, \mathbf{B}$ with characteristic polynomials $a(x), b(x)$, respectively, and for every $y > 0$, it holds that*

$$\mathcal{K}_{a \boxplus b}(y) \leq \mathcal{K}_a(y) + \mathcal{K}_b(y) - \frac{1}{y}.$$

It is worth noting that in free probability, the analog statement to Theorem 9 in the infinite dimensional case holds with equality.

3.2.2 Proof sketch of Theorem 3

The main idea in proving Theorem 3 is to randomly construct an H-local graph in a very intuitive way: since we want each vertex v to appear in d instances of H, we place v in d such instances, choosing the neighbors uniformly at random. Formally, we define a matrix $\mathcal{H}$ consisting of disjoint copies of H, and sum d random permutations of this matrix. This results in the matrix

$$X_{\mathbf{P}}(H) = \sum_{i=1}^{d} \mathbf{P}_i \mathcal{H} \mathbf{P}_i^{\mathsf{T}},$$

where $\mathbf{P} = \mathbf{P}_1, \ldots, \mathbf{P}_d$ are permutation matrices. The proof for the existence of an H-local graph whose second largest eigenvalue meet our lower bound incorporates two parts:

1. Bounding the second largest root of the *expected characteristic polynomial*, $\mathbf{E}_{\mathbf{P}}\, \chi_x\left(X_{\mathbf{P}}(H)\right)$, where the permutations are sampled uniformly and independently at random.
2. Relating the roots of $\mathbf{E}_{\mathbf{P}}\, \chi_x\left(X_{\mathbf{P}}(H)\right)$ to roots of a particular choice for $\mathbf{P}$, resulting in finding a *good* permutation, which induces a graph.

On first sight, it is not clear how to relate the expectation polynomial from Part 1 with the aforementioned tools of finite free probability. Definition 8 uses an expectation over the Haar measure, and together with Theorem 9 (applied $d-1$ times) enables us to establish an upper bound on the largest root of the expected characteristic polynomial, derived as the "free sum" of d of identical matrices. More precisely, a corollary from Theorem 9, together with the above discussion, yields

$$\mathsf{maxroot}\left(\chi_x(\mathbf{A})^{\boxplus d}\right) \leq \min_{x > \lambda_1}\left(d \cdot x - \frac{d-1}{\mathcal{G}_{\mathbf{A}}(x)}\right), \tag{3.6}$$

where λ_1 is the largest eigenvalue of $\mathbf{A}$. When taking $\mathbf{A}$ to be the matrix $\mathcal{H}$, the RHS of Equation (3.6) resembles Equation (2.6) from Theorem 2. However, the relevance of the LHS of the above equation remains unclear, as the expectation hidden in the $\boxplus$ operation is over the Haar measure and not over permutations. For bridging this gap, as well as for establishing Part 2, we follow MSS and proceed in two steps: quadrature and interlacing. While our proof makes a black-box use of these techniques we believe that the unfamiliar reader will benefit from this short account.

3.2.2.1 Quadrature

Quadrature refers to a general technique by which an integral is written as a finite sum. In our context, we make use of the result showing that finite free additive convolution can be expressed using the finite subgroup of permutation matrices of the unitary group. Specifically,

▶ **Theorem 10** (Theorem 4.1 in [22]). *Let* $\mathbf{A}, \mathbf{B}$ *be symmetric* $d \times d$ *matrices with* $\mathbf{A1} = a\mathbf{1}$ *and* $\mathbf{B1} = b\mathbf{1}$*. Let* $\chi_x(\mathbf{A}) = (x-a)a(x)$ *and* $\chi_x(\mathbf{B}) = (x-b)b(x)$*. Then,*

$$\mathop{\mathbf{E}}_{\mathbf{P}} \chi_x\left(\mathbf{A} + \mathbf{PBP}^{\mathsf{T}}\right) = (x - (a+b)) \cdot a(x) \boxplus_{d-1} b(x),$$

where $\mathbf{P}$ *is a uniformly random permutation matrix.*

Assume that $\mathbf{A}$ is the adjacency matrix of a c-regular graph, and let $p(x) = \frac{\chi_x(\mathbf{A})}{x-c}$. As a direct corollary of Theorem 10, we get that

$$\mathop{\mathbf{E}}_{\mathbf{P}_1,\ldots,\mathbf{P}_d} \chi_x\left(\sum_{i=1}^{d} \mathbf{P}_i \mathbf{A} \mathbf{P}_i^{\mathsf{T}}\right) = (x - dc) \cdot p^{\boxplus d}(x).$$

Consequently, by taking $\mathbf{A} = \mathcal{H}$, the upper bound on the max-root of the convolution polynomial, derived using Equation (3.6), directly establishes a bound on the max-root of the expected characteristic polynomial appearing on the LHS. The advantage is that we are now considering an expectation over a finite distribution, and more importantly, each element in the support of the distribution is an H-local graph. The final step of interlacing permits us to conclude that an element exists within this distribution for which the same upper bound holds.

3.2.2.2 Interlacing

So far, we have discussed how to obtain a bound on the largest root of the expected characteristic polynomial (excluding the trivial root), where the expectation is over the group of permutation matrices. It is generally incorrect to assert that a bound on the largest root of the expectation of polynomials can be utilized to infer a bound on the largest root of one of the polynomials involved in the expectation. A key observation by MSS concerning this issue is that such an result holds if the polynomials participating in the expectation form an *interlacing family*. In fact, for any choice of k, this structure suffices to deduce a bound on the k-th largest root of at least one polynomial in the family, given that we are able to bound the k-th largest root of the expected characteristic polynomial.

4 Derandomized Squaring with Interesting Graphs

In this section we apply our results to some interesting graph families, starting with graphs which demonstrate the tightness of our lower bound (see Section 4.1). Our general bound on bounded-degree graphs is given in Section 4.2 and the stronger bound assuming good spectral expansion is given in Section 4.3. The complete bipartite graph is studied in Section 4.4. The remaining sections deal with Paley graphs (see Section 4.5) and, more generally, strongly regular graphs (see Section 4.6), as well as some specific interesting graphs in Section 4.7. Some of the proofs of the results are omitted and can be found in the full version of the paper.

4.1 Three tight examples

In this short section we analyze some basic choices for the graph H, for which we know what behavior to expect and compare against our bound. We prove that for these instances, our lower bound given by Theorem 2 is tight, up to an additive vanishing term, for every Ramanujan graph G.

4.1.1 The true square

Let G be a d-regular graph, where $d \geq 3$, and let J_d denote the complete graph on d vertices, self-loops included. That is, J_d is the graph whose adjacency matrix is the all-ones $d \times d$ matrix, typically denoted as $\mathbf{J}$. Note that $G \circledS \mathrm{J}_d = G^2$. Our lower bound on $\lambda(G \circledS \mathrm{J}_d)$ obtained in Theorem 2 holds for all graphs G in the sense that Λ_{J_d} is independent of G. Therefore, it is sensible to expect that if the lower bound is tight for this choice of $H = \mathrm{J}_d$ then it is matched by taking G to be a d-regular Ramanujan graph. In such case, the spectral expansion $\lambda(G \circledS \mathrm{J}_d) = \lambda(G^2)$ can be computed directly and is equal to $\lambda(G)^2 = \left(2\sqrt{d-1}\right)^2 = 4(d-1)$. As we will now show, it is indeed the case that $\Lambda_{\mathrm{J}_d} = 4(d-1)$, establishing that our lower bound is tight for this choice of H.

The Cauchy transform of J_d is given by

$$\mathcal{G}_{\mathrm{J}_d}(x) = \frac{1}{d}\left(\frac{d-1}{x} + \frac{1}{x-d}\right) = \frac{x-d+1}{x(x-d)}.$$

Substituting to Equation (2.5) and simplifying, we see that the derandomized squaring polynomial associated with J_d is given by $\Delta_{\mathrm{J}_d}(x) = x^2 + (2-2d)x$. The unique real positive root of $\Delta_{\mathrm{J}_d}(x)$ is, of course $x_0 = 2d-2$. Substituting to Equation (2.6) yields $\Lambda_{\mathrm{J}_d} = 4(d-1)$.

4.1.2 Non-backtracking random walks

A second example for the tightness of our bound is given by the clique, denoted as K_d, namely the complete graph without self-loops. It is easily seen combinatorially that $G \circledS \mathrm{K}_d$ corresponds to the graph of non-backtracking length-2 walks in G, which we denote by $G^{(2)}$. The corresponding adjacency matrix is given by $\mathbf{A}^{(2)} = \mathbf{A}^2 - d\mathbf{I}$. Hence, for a Ramanujan graph G, the best value we can hope for coming from the analysis is

$$\lambda(\mathbf{A}^{(2)}) = \lambda(\mathbf{A}^2) - d = 4(d-1) - d = 3d - 4.$$

This bound is indeed what results from our analysis. To see this, note that the Cauchy transform of K_d is

$$\mathcal{G}_{\mathrm{K}_d}(x) = \frac{1}{d}\left(\frac{d-1}{x+1} + \frac{1}{x-d+1}\right) = \frac{x-d+2}{(x+1)(x-d+1)}.$$

Substituting to Equation (2.5) and simplifying, we see that the derandomized squaring polynomial associated with K_d is given by

$$\Delta_{\mathrm{K}_d}(x) = x^2 + (4-2d)x + 3 - 2d.$$

The unique real positive root of $\Delta_{\mathrm{K}_d}(x)$ is $x_0 = 2d-3$. Substituting to Equation (2.6) yields the desired bound, $\Lambda_{\mathrm{K}_d} = 3d-4$.

It is instructive to compare our bound with the upper bound obtained by the Rozenman-Vadhan bound, Equation (1.1) for a Ramanujan graph G. The second largest (normalized) eigenvalue of K_d in absolute value is $\frac{1}{d-1}$, leading to an overall bound of $\approx 5d$ compared to the true behavior of $\approx 3d$.

4.1.3 The graph achieving the Rozenman-Vadhan bound

In [35], Rozenman and Vadhan proved that the bound that is given in Equation (1.1) is tight in a strong sense, namely, for every rational μ, there exists a graph RV_μ with $\omega(\mathrm{RV}_\mu) = \mu$, such that for every graph G it holds that $\omega(G \circledS \mathrm{RV}_\mu) = (1-\mu)\omega(G^2) + \mu$. For ease of

notation, we write RV instead of RV_μ from hereon. The graph RV is constructed as follows: assume $\mu = \frac{s}{r}$ for some integers $r, s \geq 1$, then RV is a graph on d vertices which is rd-regular, consisting of $r - s$ copies of the complete graph J_d, and additional sd self-loops on each vertex. The combinatorial analysis shows that a step in the rd^2-regular graph $G \circledS \mathrm{RV}$ amounts to staying at the same vertex with probability μ or taking a step in G^2 with probability $1 - \mu$, leading directly to the tightness of the bound.

Assume once again that G is Ramanujan, hence $\lambda(G) \leq 2\sqrt{d-1}$. In this case, by Equation (1.1), we get that

$$\begin{aligned}\lambda(G \circledS \mathrm{RV}) &= \omega(G \circledS \mathrm{RV}) \cdot rd^2 \\ &= \left(\left(1 - \frac{s}{r}\right)\frac{4(d-1)}{d^2} + \frac{s}{r}\right) \cdot rd^2 \\ &= 4(d-1)r + (d-2)^2 s.\end{aligned}$$

As in the above two examples, we use the Cauchy transform of RV which is given by

$$\mathcal{G}_{\mathrm{RV}}(x) = \frac{1}{d}\left(\frac{d-1}{x - sd} + \frac{1}{x - rd}\right) = \frac{x - (d-1)r - s}{(x - dr)(x - ds)}.$$

Expanding Equation (2.7) and solving, we get

$$\Lambda_{\mathrm{RV}} = \psi_{\mathrm{RV}}(x_0) = 4(d-1)r + (d-2)^2 s,$$

as in the Rozenman-Vadhan bound.

4.2 Bounded-degree graphs and a universal bound on κ

In this section we prove that derandomized squaring with a simple vertex-transitive graph of bounded degree results with a graph that gravitates towards Ramanujan as the number of vertices in H increases. Moreover, we establish a universal bound on κ_H which holds for all simple vertex-transitive graphs.

▶ **Theorem 11.** *Let H be a simple vertex-transitive c-regular graph on d vertices, where $d \geq 3$ and $c \geq 1$. Then,*

$$\kappa_H \leq 2 + \frac{\sqrt{c}}{\sqrt{d} - \sqrt{c}}.$$

Moreover, for every $d \geq 11$ it holds that $\kappa_H \leq 3$. Lastly, if H is triangle-free then

$$\kappa_H \leq 2 + \frac{c}{d} \cdot \frac{1}{1 - \sqrt{\frac{c}{d}}}.$$

4.3 Spectral expanders

In this section we prove a stronger bound on κ_H than the one obtained in Theorem 11, which recall holds for general bounded-degree graphs, assuming that H is a good spectral expander. The proof follows the same argument as in the proof of Theorem 11 though takes into account the bound on the spectral expansion.

▶ **Proposition 12.** *Let H be a simple vertex-transitive c-regular graph on d vertices. Assume that $10 \leq c < \sqrt{d}$. Then,*

$$\kappa_H \leq 2 + \frac{3}{\sqrt{d}}\left(\frac{\lambda(H)}{\sqrt{c}}\right)^3.$$

Furthermore, if H is triangle-free then

$$\kappa_H \leq 2 + \frac{4}{d}\left(\frac{\lambda(H)}{\sqrt{c}}\right)^4.$$

4.4 Complete bipartite graphs

In this section we consider the operation of derandomized squaring with the complete bipartite graph on d vertices, for an even integer $d \geq 4$. This is, of course, a $c = \frac{d}{2}$ regular graph which we denote as $\mathrm{K}_{c,c}$. It is easy to see that $\mathrm{K}_{c,c}$ has eigenvalues $\pm c$, each with multiplicity 1, and the remaining eigenvalues are all 0. Therefore, the corresponding Cauchy transform, which for ease of notation we denote by $\mathcal{G}_{c,c}$, is given by

$$\mathcal{G}_{c,c}(x) = \frac{4x^2 - d(d-2)}{(4x^2 - d^2)x}.$$

From here one can compute the derandomized squaring polynomial

$$\Delta_{c,c}(x) = x^4 - \frac{d(2d-3)}{2}x^2 - \frac{(d-2)d^3}{16}$$

whose unique positive root $x_0 = \frac{\sqrt{md}}{2}$, where $m = 2d - 3 + \sqrt{(d-1)(5d-9)}$. One can verify that $\mathcal{G}(x_0) = \frac{m-d+2}{x_0(m-d)}$. Substituting this to Equation (2.6), we get

$$\Lambda_{\mathrm{K}_{c,c}} = x_0\left(\frac{m+d}{m-d+2}\right) = \frac{\sqrt{md}}{2}\left(\frac{m+d}{m-d+2}\right).$$

The resulted graph is $D = \frac{d^2}{2}$-regular, and so we normalize by dividing by $\sqrt{D-1}$ to get

$$\kappa_{\mathrm{K}_{c,c}} = \sqrt{\frac{md}{2d^2-4}}\left(\frac{m+d}{m-d+2}\right).$$

From here one can compute the first couple of values, $\kappa_{\mathrm{K}_{2,2}} \approx 2.089$ which, of course, matches the bound we computed for the length-4 cycle, and $\kappa_{\mathrm{K}_{3,3}} \approx 2.157$. Considering the limit behavior as $d \to \infty$ we have that $m \approx \gamma d$ where $\gamma = 2 + \sqrt{5}$, and so

$$\kappa_{\mathrm{K}_{\infty,\infty}} \triangleq \lim_{c\to\infty} \kappa_{\mathrm{K}_{c,c}} = \sqrt{\frac{\gamma}{2}} \cdot \frac{\gamma+1}{\gamma-1} = \frac{1}{2}\sqrt{11 + 5\sqrt{5}} \approx 2.355.$$

4.5 Paley graphs

Let $q = 4r + 1$ be a prime power. The *Paley graph*, denoted as Pal_q, has vertex set corresponding to the finite field $\mathbb{F}_q$ with the vertices adjacent if and only if their difference is a nonzero square in $\mathbb{F}_q$. As $q \equiv 1$ modulo 4, we have that -1 is a square in $\mathbb{F}_q$, and so Pal_q is undirected.

▶ **Theorem 13.**

$$\kappa_{\mathrm{Pal}_\infty} \triangleq \lim_{q\to\infty} \kappa_{\mathrm{Pal}_q} = \frac{1 + \sqrt{13 + 16\sqrt{2}}}{\sqrt{8}} \approx 2.464.$$

4.6 Strongly regular graphs

A c-regular graph on d vertices with no self-loops is called *strongly regular* with parameters λ, μ [4] if $0 < c < d$ (namely, the graph is neither complete nor edgeless) and the following hold:

1. For each pair of adjacent vertices there are λ vertices adjacent to both.
2. For each pair of nonadjacent vertices there are μ vertices adjacent to both.

Strongly regular graphs include dozens of interesting graphs such as the Peterson graph, the Hoffman-Singelton graph (see Section 4.7), and the symplectic graphs (see Section 4.6.1), as well as all Paley graphs which we analyzed in Section 4.5. It is a well-known fact that strongly regular graphs have two eigenvalues other than the trivial eigenvalue c, denoted r, s, with the convention $r > s$. This is, in fact, a spectral characterization of strongly regular graph among regular graphs. The multiplicity of these eigenvalues are denoted by f, g, respectively. In the following we compute the derandomized squaring polynomial of a strongly regular graph.

▶ **Proposition 14.** *Let H be a c-regular strongly regular graph on $d \geq 3$ vertices with parameters λ, μ. Let $\alpha = \lambda - \mu$ and $e = c - d + 1$. Then, the derandomized squaring polynomial associated with H is given by*

$$\Delta_H(x) = x^4 - 2(\alpha + c)x^3 + Ax^2 + Bx + C,$$

where

$$\begin{aligned} A &= 2\mu + 4\alpha c + \alpha^2 + ce, \\ B &= -2\alpha\mu + 2c\left(c - ce - \alpha^2 - \mu\right), \\ C &= (\mu - c)\left(\mu + 2\alpha c + c\right) + c^3 e. \end{aligned}$$

4.6.1 The symplectic graphs

An interesting sub-family of strongly regular graphs are the so-called *symplectic graphs*. Let $r \geq 1$ be an integer. The symplectic graph $\mathrm{Sp}(2r)$ is the graph whose vertex set consists of all nonzero vectors in $\mathbb{F}_2^{2r}$. Two vertices x, y are adjacent whenever $x^{\mathsf{T}}\mathbf{N}y = 1$, with all calculations over $\mathbb{F}_2$, where $\mathbf{N}$ is the $(2r) \times (2r)$ block diagonal matrix with r blocks of the form $\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$. It is well-known that $\mathrm{Sp}(2r)$ is a strongly regular graph with $\lambda = \mu = 2^{2r-2}$.

▶ **Theorem 15.**

$$\lim_{r\to\infty} \kappa(\mathrm{Sp}(2r)) = \frac{1 + \sqrt{13 + 16\sqrt{2}}}{\sqrt{8}} \approx 2.464.$$

Interestingly, though not unexpectedly, this limit behavior is also shared by Paley graphs (see Section 4.5).

4.7 Some specific graphs

In this section we consider some specific interesting graphs.

[4] It is customary to denote these parameters by λ and μ and so, despite our use of λ for denoting the spectral expansion, we proceed with this standard notation.

Table 1 Summary of specific graphs and their derandomized squaring constants.

H (Graph name)	d (Number of vertices)	c (degree)	κ_H
Petersen graph	10	3	2.025
Heawood graph	14	3	2.004
Hoffman-Singelton graph	50	7	2.007
Biggs-Smith graph	102	3	2.000000016
Conway's 99-graph	99	14	2.041

References

1 AmirMahdi Ahmadinejad, Jonathan Kelner, Jack Murtagh, John Peebles, Aaron Sidford, and Salil Vadhan. High-precision estimation of random walks in small space. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science*, pages 1295–1306. IEEE Computer Soc., Los Alamitos, CA, [2020] ©2020. doi:10.1109/FOCS46700.2020.00123.

2 AmirMahdi Ahmadinejad, John Peebles, Edward Pyne, Aaron Sidford, and Salil Vadhan. Singular value approximation and reducing directed to undirected graph sparsification. *arXiv preprint arXiv:2301.13541*, 2023. doi:10.48550/arXiv.2301.13541.

3 N. Alon and F. R. K. Chung. Explicit construction of linear sized tolerant networks. In *Proceedings of the First Japan Conference on Graph Theory and Applications (Hakone, 1986)*, volume 72, pages 15–19, 1988. doi:10.1016/0012-365X(88)90189-6.

4 Noga Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986. doi:10.1007/BF02579166.

5 Avraham Ben-Aroya and Amnon Ta-Shma. A combinatorial construction of almost-Ramanujan graphs using the zig-zag product. *SIAM J. Comput.*, 40(2):267–290, 2011. doi:10.1137/080732651.

6 Yonatan Bilu and Nathan Linial. Lifts, discrepancy and nearly optimal spectral gap. *Combinatorica*, 26(5):495–519, 2006. doi:10.1007/s00493-006-0029-7.

7 Charles Bordenave. A new proof of Friedman's second eigenvalue theorem and its extension to random lifts. *Ann. Sci. Éc. Norm. Supér. (4)*, 53(6):1393–1439, 2020. doi:10.24033/asens.2450.

8 Charles Bordenave and Benoît Collins. Eigenvalues of random lifts and polynomials of random permutation matrices. *Annals of Mathematics*, 190(3):811–875, 2019.

9 Mark Braverman, Gil Cohen, and Sumegha Garg. Pseudorandom pseudo-distributions with near-optimal error for read-once branching programs. *SIAM Journal on Computing*, 49(5):STOC18–242, 2019. doi:10.1137/18M1197734.

10 Eshan Chattopadhyay and Jyun-Jie Liao. Optimal error pseudodistributions for read-once branching programs. In *35th Computational Complexity Conference*, volume 169 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 25, 27. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.CCC.2020.25.

11 Lijie Chen, William Hoza, Xin Lyu, Avishay Tal, and Hongxun Wu. Weighted pseudorandom generators via inverse analysis of random walks and shortcutting. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 114, 2023.

12 Gil Cohen and Gal Maor. Random walks on rotating expanders. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 971–984, 2023. doi:10.1145/3564246.3585133.

13 Michael B. Cohen. Ramanujan graphs in polynomial time. In *57th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2016*, pages 276–281. IEEE Computer Soc., Los Alamitos, CA, 2016. doi:10.1109/FOCS.2016.37.

14 Joel Friedman. A proof of Alon's second eigenvalue conjecture and related problems. *Mem. Amer. Math. Soc.*, 195(910):viii+100, 2008. doi:10.1090/memo/0910.

15 Rostislav I Grigorchuk and Andrzej Zuk. On the asymptotic spectrum of random walks on infinite families of graphs. *Random walks and discrete potential theory (Cortona, 1997)*, pages 188–204, 1999.

16 Chris Hall, Doron Puder, and William F. Sawin. Ramanujan coverings of graphs. *Adv. Math.*, 323:367–410, 2018. doi:10.1016/j.aim.2017.10.042.

17 Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

18 Yasutaka Ihara. On discrete subgroups of the two by two projective linear group over p-adic fields. *J. Math. Soc. Japan*, 18:219–235, 1966. doi:10.2969/jmsj/01830219.

19 Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988. doi:10.1007/BF02126799.

20 Adam W. Marcus, Daniel A. Spielman, and Nikhil Srivastava. Interlacing families I: Bipartite Ramanujan graphs of all degrees. *Ann. of Math. (2)*, 182(1):307–325, 2015. doi:10.4007/annals.2015.182.1.7.

21 Adam W. Marcus, Daniel A. Spielman, and Nikhil Srivastava. Interlacing families II: Mixed characteristic polynomials and the Kadison-Singer problem. *Ann. of Math. (2)*, 182(1):327–350, 2015. doi:10.4007/annals.2015.182.1.8.

22 Adam W Marcus, Daniel A Spielman, and Nikhil Srivastava. Interlacing families IV: Bipartite Ramanujan graphs of all sizes. *SIAM Journal on Computing*, 47(6):2488–2509, 2018. doi:10.1137/16M106176X.

23 Adam W. Marcus, Daniel A. Spielman, and Nikhil Srivastava. Finite free convolutions of polynomials. *Probab. Theory Related Fields*, 182(3-4):807–848, 2022. doi:10.1007/s00440-021-01105-w.

24 Grigorii Aleksandrovich Margulis. Explicit group-theoretical constructions of combinatorial schemes and their application to the design of expanders and concentrators. *Problemy peredachi informatsii*, 24(1):51–60, 1988.

25 Sidhanth Mohanty and Ryan O'Donnell. X-Ramanujan graphs. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms*, pages 1226–1243. SIAM, Philadelphia, PA, 2020. doi:10.1137/1.9781611975994.75.

26 Sidhanth Mohanty, Ryan O'Donnell, and Pedro Paredes. Explicit near-Ramanujan graphs of every degree. *SIAM J. Comput.*, 51(3):1–23, 2022. doi:10.1137/20M1342112.

27 Moshe Morgenstern. Existence and explicit constructions of $q + 1$ regular Ramanujan graphs for every prime power q. *J. Combin. Theory Ser. B*, 62(1):44–62, 1994. doi:10.1006/jctb.1994.1054.

28 Jack Murtagh, Omer Reingold, Aaron Sidford, and Salil Vadhan. Derandomization beyond connectivity: undirected Laplacian systems in nearly logarithmic space. *SIAM J. Comput.*, 50(6):1892–1922, 2021. doi:10.1137/20M134109X.

29 Alexandru Nica and Roland Speicher. *Lectures on the combinatorics of free probability*, volume 335 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 2006. doi:10.1017/CBO9780511735127.

30 A. Nilli. On the second eigenvalue of a graph. *Discrete Math.*, 91(2):207–210, 1991. doi:10.1016/0012-365X(91)90112-F.

31 Ryan O'Donnell and Xinyu Wu. Explicit near-fully X-Ramanujan graphs. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science*, pages 1045–1056. IEEE Computer Soc., Los Alamitos, CA, 2020.

32 Karol A Penson and Karol Życzkowski. Product of ginibre matrices: Fuss-Catalan and Raney distributions. *Physical Review E*, 83(6):061118, 2011.

33 Omer Reingold. Undirected connectivity in log-space. *J. ACM*, 55(4):Art. 17, 24, 2008. doi:10.1145/1391289.1391291.

34 Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 3–13. IEEE, 2000. doi:10.1109/SFCS.2000.892006.

35 Eyal Rozenman and Salil Vadhan. Derandomized squaring of graphs. In *Approximation, randomization and combinatorial optimization*, volume 3624 of *Lecture Notes in Comput. Sci.*, pages 436–447. Springer, Berlin, 2005. doi:10.1007/11538462_37.

36 Amnon Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 238–251, 2017. doi:10.1145/3055399.3055408.

37 Wolfgang Woess. *Random walks on infinite graphs and groups*, volume 138 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 2000. doi:10.1017/CBO9780511470967.

Query Complexity of Stochastic Minimum Vertex Cover

Mahsa Derakhshan
Northeastern University, Boston, MA, USA

Mohammad Saneian
Northeastern University, Boston, MA, USA

Zhiyang Xun
University of Texas at Austin, TX, USA

Abstract

We study the stochastic minimum vertex cover problem for general graphs. In this problem, we are given a graph $G = (V, E)$ and an existence probability p_e for each edge $e \in E$. Edges of G are realized (or exist) independently with these probabilities, forming the realized subgraph $\mathcal{G}$. The existence of an edge in $\mathcal{G}$ can only be verified using edge queries. The goal of this problem is to find a near-optimal vertex cover of $\mathcal{G}$ using a small number of queries.

Previous work by Derakhshan, Durvasula, and Haghtalab [STOC 2023] established the existence of $1.5 + \varepsilon$ approximation algorithms for this problem with $O(n/\varepsilon)$ queries. They also show that, under mild correlation among edge realizations, beating this approximation ratio requires querying a subgraph of size $\Omega(n \cdot \mathrm{RS}(n))$. Here, $\mathrm{RS}(n)$ refers to Ruzsa-Szemerédi Graphs and represents the largest number of induced edge-disjoint matchings of size $\Theta(n)$ in an n-vertex graph.

In this work, we design a simple algorithm for finding a $(1 + \varepsilon)$ approximate vertex cover by querying a subgraph of size $O(n \cdot \mathrm{RS}(n))$ for any absolute constant $\varepsilon > 0$. Our algorithm can tolerate up to $O(n \cdot \mathrm{RS}(n))$ correlated edges, hence effectively completing our understanding of the problem under mild correlation.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Sublinear algorithms, Vertex cover, Query complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.41

Funding *Zhiyang Xun*: Supported by NSF Grant CCF-2312573 and a Simons Investigator Award (#409864, David Zuckerman).

Acknowledgements The authors thank Sanjeev Khanna for countless insightful discussions and help.

1 Introduction

We study the *stochastic minimum vertex cover* problem. In this problem, we are given an arbitrary n-vertex graph $G(V, E)$ and an existence probability p_e for each edge $e \in E$. Edges of G are realized or exist independently, with their corresponding probability forming the realized subgraph $\mathcal{G}$. While $\mathcal{G}$ is unknown, one can query an edge $e \in E$ to see if this edge exists in $\mathcal{G}$. The goal is to find a near-optimal vertex cover of $\mathcal{G}$ by querying a *small* subset of E.

There is extensive literature studying various graph problems in this stochastic setting, such as minimum spanning tree [21, 22], all-pairs shortest paths [24], and maximum matching [13, 5, 12, 25, 3, 6, 9, 10, 8, 16, 15]. The stochastic vertex cover for general graphs was first studied by Blum, Behnezhad, and Derakhshan [7]. They provided a 2-approximation algorithm with $O(n/p)$ queries where $p = \min_{e \in E} p_e$. Later, this approximation ratio was improved to $1.5 + \varepsilon$ by the work of Derakhshan, Durvasula, and Haghtalab [14] with $O(n/\varepsilon p)$


© Mahsa Derakhshan, Mohammad Saneian, and Zhiyang Xun;
licensed under Creative Commons License CC-BY 4.0



queries. Even though these stochastic settings are primarily concerned with information-theoretical questions, this was the first work to provide positive results for a computationally hard problem. However, the key unresolved question remains:

▶ **Question 1.** *How many non-adaptive edge queries are needed to find a* $(1+\varepsilon)$*-approximation to the minimum vertex cover problem in the stochastic setting?*

In this paper, we take a step toward answering this question. We design a simple algorithm and establish a relation between the number of queries it makes and the density of a well-known class of graphs called Ruzsa-Szemerédi Graphs (Definition 2). Our number of queries matches the lower bound of Derakhshan, Haghtalab, and Durvasula [14] for the number of queries needed to surpass a 1.5-approximation in the presence of $O(n)$ correlated edges.

▶ **Definition 2** (Ruzsa-Szemerédi Graphs [23])**.** *An n-vertex graph $G(V,E)$ is an $RS_n(r,t)$ graph if its edge set E can be decomposed into t edge-disjoint induced matchings of size r. We use $RS_n(r)$ to denote the maximum t for which $RS_n(r,t)$ exists.*

Our main result is formalized in the theorem below.

▶ **Result 3.** *Given any $\varepsilon \in (0,1)$ let t be $RS_n(cn)$ for $c = \frac{\varepsilon^2}{128}$. There exists an algorithm that queries a subgraph of size $O(tn/p)$ and finds a vertex cover with the expected approximation ratio of $(1+\varepsilon)$.*

Proving lower and upper bounds parameterized by the density of RS-graphs has been of interest in various communities, such as dynamic algorithms [11, 4], streaming algorithms [20, 1, 2], property testing [17], and additive combinatorics [19]. Despite this, our understanding of their density remains incomplete. What we currently know is:

$$n^{\Omega_c(1/\log\log(n))} \overset{[17]}{\leq} RS_n(cn) \overset{[18]}{\leq} n/\log^{(O(\log(1/c)))}$$

where $\log^{(x)}$ is the x-iterated logarithmic function. Our result is most interesting if the lower-bound turns out to be tight. In such a case the number of edges queried by our algorithms is

$$|Q| = O(n^{1+\Omega_c(1/\log\log(n))}/p) = O(n^{1+o(1)}/p).$$

On the opposite end, if the actual density of RSgraphs is closer to the known upper-bound, our result implies $|Q| = O(\frac{n^2}{p\log^{O(\log(1/c))}})$.

1.1 Technical Overview

We first design an algorithm with an additive error of at most εn in Section 2 and later show how it can be modified to obtain the $(1+\varepsilon)$ approximation ratio. To choose which edges to query we rely on a parameter c_e defined for any edge $e \in E$ as follows:

$$c_e = \Pr[e \text{ has an end-point in OPT}],$$

where $\text{OPT} = \text{MVC}(\mathcal{G})$. In this definition, the randomization comes from the realization of $\mathcal{G}$ and MVC(.) is a fixed algorithm which returns the minimum vertex cover of a given graph. Basically, c_e is the probability with which an edge $e \in E$ is covered by $\text{MVC}(\mathcal{G})$.[1]

[1] Since we are not concerned with computational efficiency, we assume c_es are known. Regardless, similar to [14], our results can also be obtained with the estimated value of c_es calculated using polynomially many calls to the MVC oracle.

Given a parameter $\tau \in (0,1)$ whose value we fix later, we set Q, the subgraph that we query, to be the subgraph of edges with $c_e \leq 1-\tau$. I.e.,

$$Q = \{e \in E : c_e \leq 1-\tau\}.$$

Since we need the output to be a valid vertex cover, our final solution also needs to cover all the edges that we do not query. Let us denote this subgraph by $H = G - Q$. Therefore, after we query Q and see its realization $\mathcal{Q}$, the most straightforward way of finding a small vertex cover is to output $\text{MVC}(\mathcal{Q} \cup H)$. However, for the sake of analysis, we need to design an alternative algorithm for finding a vertex cover of $\mathcal{Q} \cup H$. We formally explain this in Algorithm 1 and briefly explain the overall ideas below.

After querying Q and observing its realization $\mathcal{Q}$, we also hallucinate the edges in H to get $\mathcal{H}$. The hallucinated subgraph $\mathcal{H}$ is a random realization of H containing each edge independently w.p. p_e. We then let $P_1 = \text{MVC}(\mathcal{Q} \cup \mathcal{H})$ and commit all its vertices to the final solution. Next, we pick another set of vertices, P_2, to cover all the edges in H that were not covered by P_1. That is, $P_2 = \text{MVC}(H[V - P_1])$. We finally output $P_1 \cup P_2$. This clearly covers both $\mathcal{Q}$ and H and is a valid solution. The extra step of hallucinating H is to ensure that P_1 has the same distribution as the optimal solution (see Claim 8). This gives us two properties:

1. The expected size of P_1 is the same as OPT. To analyze our approximation ratio, it suffices to upper-bound $\mathbf{E}[|P_2|]$.
2. For any edge $e \in H$, we have $\Pr[e \text{ not covered by } P_1] = 1 - c_e$. Since by our choice of Q, the edges in $H = G - Q$ all have $c_e \geq 1 - \tau$. This implies that any edge in H remains in $H[V - P_1]$ w.p. at most τ.

Having this, we argue that if the expected size of the solution $P_1 \cup P_2$ is more than $\mathbf{E}[\text{OPT}] + \varepsilon n$, then we have $|P_2| \geq \varepsilon n$. This means that the subgraph $R = H[V - P_1]$ has a large minimum vertex cover and, consequently, a large maximum matching. Now, suppose we draw T independent subgraphs of H from the same distribution as R and call them $R_1, \ldots, R_T$. (We can do so by running our algorithm on T different hallucinations of G (see Definition 4).)

From each R_i, we take the maximum matching of it to be M_i, and from M_i, we delete all the edges present in at least one other R_j for $j \neq i$. We show that this gives us a set of induced matchings. Notice that since we have that any edge is present in R_i with probability at most τ, we can argue that deleting the edges present in other R_j's will not reduce the size of M_i significantly for a choice of $T = \frac{1}{2\tau}$. Therefore, we conclude that if $\mathbf{E}[|P_2|]$ is greater than εn, we can find a dense RS-graph in H. Finally, by carefully choosing $\tau = 1/3t$, where t is $\text{RS}_n(\varepsilon n/8)$, we argue that Algorithm 1 has an additive approximation of at most εn.

In Section 3, we explain how we can modify this algorithm and design our $(1+\varepsilon)$ algorithm Algorithm 3. The main difference is that we pre-commit a subset of vertices to the final solution at the beginning of the algorithm. These vertices are the ones that have a high chance of being in OPT; therefore, including them does not significantly increase the size of the output. This allows us to argue that the optimal solution on H is large enough to allow us to turn the additive error into a multiplicative error.

1.2 Preliminaries

Throughout the paper, we denote the input graph as $G = (V, E)$, where $|V| = n$. From this input graph, we have each edge e present in $\mathcal{G}$ independently with probability p_e. We denote the minimum value among all p_es to be p. Given a small constant $\varepsilon > 0$, our goal is to query a sparse subgraph of G represented by Q non-adaptively and find a $(1+\varepsilon)$ approximate minimum vertex cover of $\mathcal{G}$.

We let MVC() be a fixed algorithm which given any graph outputs its minimum vertex cover. We may also refer to this as the minimum vertex cover oracle. We define $\text{OPT} = \text{MVC}(\mathcal{G})$ to be the optimal solution to our problem. Note that OPT is a random variable since $\mathcal{G}$ is a random realization of G. We also let $\textsc{opt} = \mathbf{E}[|\text{OPT}|]$ be the expected size of this optimal solution. We let $\mu(G)$ be the size of the maximum matching of the graph G.

For any vertex $v \in V$, we define

$$c_v = \Pr[v \in \text{OPT}],$$

which is the probability that v joins the optimal solution. This means $\sum_{v \in V} c_v = \textsc{opt}$. Similarly for any edge $e = (u, v)$ in the graph G we define

$$c_e = \Pr[u \in \text{OPT or } v \in \text{OPT}].$$

▶ **Definition 4** (Graph Hallucination). *Given a any subgraph L of G we define a hallucination of L to be a subgraph $\mathcal{L}$ which contains each edge $e \in L$ independently with probability p_e.*

We define $\text{RS}(n)$ to be the largest number of induced edge-disjoint matchings of size $\Theta(n)$ in an n-vertex graph.

2 The algorithm

In this section, we prove the following theorem.

▶ **Theorem 5.** *Let t be $RS_n(\varepsilon n/8)$. Running Algorithm 1 with $\tau = \frac{1}{3t}$ finds a vertex cover that has an additive approximation of at most εn to the optimal minimum vertex cover and queries a subgraph of size $O(tn/p)$.*

To prove this theorem, we design Algorithm 1, which takes a parameter τ, and queries the subgraph Q with $c_e \leq 1 - \tau$. Let us call the edges realized in Q to be $\mathcal{Q}$. We then hallucinate the edges not in Q which we call the subgraph $H = G - Q$ to get $\mathcal{H}$. We then let P_1 be $\text{MVC}(\mathcal{Q} \cup \mathcal{H})$.

This makes P_1 have the same distribution of OPT (see Claim 8 for the proof). Now we commit the vertices in P_1 to the final solution and all that remains to be covered are edges in H that have not been covered so far. Therefore we add $P_2 = \text{MVC}(H[V - P_1])$ to the solution. All we need to do is to argue $\mathbf{E}[|P_2|]$ is small since we have $\mathbf{E}[|P_1|] = \textsc{opt}$.

We show that if $\mathbf{E}[|P_2|]$ is large we can find a large RS-graph in H (see Algorithm 2) so by carefully choosing τ which we relate to the density of RS-graphs we prove $\mathbf{E}[|P_2|]$ is small and hence we prove Algorithm 1 has at most additive error of εn. In the next section we modify Algorithm 1 slightly to get Algorithm 3 and show that this algorithm finds a $(1 + \varepsilon)$ approximation to the minimum vertex cover.

2.1 A Two-Step Algorithm

Here we describe the algorithm for finding the minimum vertex cover of G. We call it the two step algorithm because the output consists of union of two sets P_1 and P_2.

First, let us prove that the output of Algorithm 1 is a vertex cover.

▷ Claim 6. The output of Algorithm 1 is a vertex cover for $\mathcal{G}$.

Proof. Since P_2 is a vertex cover on the graph that P_1 does not cover we can argue that $P_1 \cup P_2$ is a vertex cover for $\mathcal{G}$. ◁

Algorithm 1 An algorithm to find MVC for $\mathcal{G}$.

Parameter: τ

1. Let Q be the subgraph of edges with $c_e \leq 1 - \tau$.
2. Let H be $G - Q$.
3. Query edges in Q to get its realization $\mathcal{Q}$.
4. Hallucinate edges in H to form $\mathcal{H}$.
5. Let $P_1 = \text{MVC}(\mathcal{Q} \cup \mathcal{H})$.
6. Let $R = H[V \setminus P_1]$.
7. Let $P_2 = \text{MVC}(R)$.
8. Return $O = P_1 \cup P_2$.

Now let us bound the number of queries that Algorithm 1 makes.

▷ Claim 7. The size of subgraph Q that we query is at most $O(\frac{n}{p\tau})$ for $p = \min(p_e)$ for $e \in E$.

Proof. Due to [14] we know that if we take a random minimum vertex cover P for $\mathcal{G}$ at most $O(n/p)$ edges will not be covered by it with high probability. Here we repeat how this argument works for convenience. Let us call this number of edges X and call this subgraph F. For any subgraph with more than n/p edges, the probability of none of its edges being in $\mathcal{G}$ is at most $(1-p)^{n/p}$. Moreover, since F is an induced subgraph of G, there are at most 2^n possibilities for it. By a union bound, we get that with high probability, F has at most n/p edges. That is:

$$\Pr[|E(F)| \leq n/p] \geq 1 - 2^n(1-p)^{n/p} \geq 1 - \frac{2^n}{e^n} = 1 - (2/e)^n$$

Due to linearity of expectation we have

$$E[X] = \sum_e 1 - c_e.$$

For the edges in Q we have $1 - c_e \geq \tau$ so we can get $|E(Q)| \leq \frac{X}{\tau}$ and since we have $X = O(n/p)$ we get that $|E(Q)| \leq O(\frac{n}{p\tau})$. ◁

▷ Claim 8. The expected size of P_1 from Algorithm 1 is equal to OPT and the distribution of P_1 is the same as OPT

Proof. Look at the graph that P_1 is a minimum vertex cover for. It is the graph $\mathcal{Q} \cup \mathcal{H}$. For any edge e, if it is in H then it is present in $\mathcal{H}$ with probability p_e and if it is in Q it is present in $\mathcal{Q}$ with probability p_e. Since $Q \cup H = G$ we can see that the distribution of $\mathcal{Q} \cup \mathcal{H}$ is the same as the distribution for $\mathcal{G}$ and hence the expected size of the minimum vertex cover for $\mathcal{Q} \cup \mathcal{H}$ is the same as $\mathcal{G}$ which is OPT. Also since the distribution of $\mathcal{Q} \cup \mathcal{H}$ is the same as $\mathcal{G}$ we get

$$P_1 = \text{MVC}(\mathcal{Q} \cup \mathcal{H}) = \text{MVC}(\mathcal{G}) = \text{OPT}$$

Hence P_1 comes from the same distribution as OPT. ◁

▶ **Lemma 9.** *If Algorithm 1 does not have additive approximation of at most k in expectation, then* $\mathbf{E}[\mu(R)] \geq \frac{k}{2}$

Proof. Due to Claim 8, the expected size of P_1 is OPT. Therefore, if Algorithm 1 has worse than k additive approximation then we can conclude $\mathbf{E}[|P_2|] \geq k$. In any graph the maximum matching size it at least half of the size of minimum vertex cover. Applying this statement on the graph R give us that

$$\mathbf{E}[\mu(R)] \geq \frac{E[|P_2|]}{2} \geq \frac{k}{2}.$$ ◀

2.2 Finding a Dense RS-graph in H

Suppose in Algorithm 1 instead of querying Q and hallucinating H we hallucinate all the edges. We repeat this process T times, calling the resulting graph R in Algorithm 1 R_i for each i from 1 to T. For each R_i, we find the maximum matching M_i. Then, from each M_i, we delete the edges that are present in at least one other R_j for $j \neq i$, and we define S_i to be the remaining edges of M_i after these deletions. In Algorithm 2, we formalize this process. Our goal in this section is to argue that if Algorithm 1 does not have an additive error of at most εn, then the matchings S_i will be large, and we can find a large RS-graph as a subgraph of H.

Algorithm 2 Finding a large RS-graph in H in Algorithm 1.

Parameter: τ

1. For i from 1 to T:
   a. Let Q and H be graphs defined in Algorithm 1
   b. Hallucinate graph G to get $\mathcal{G}_i$
   c. Let P_i be MVC($\mathcal{G}_i$)
   d. Let $R_i = H[V \setminus P_i]$
2. Let M_i be the maximum matching of R_i
3. Delete the edges in M_i present in at least one other R_j with $j \neq i$ to get S_i
4. Return $S_1 \cup S_2 \ldots \cup S_T$

The next claim will bound the probability of edges being in R_i.

▷ Claim 10. For each edge e and each graph R_i, we have that $\Pr[e \in R_i] \leq \tau$.

Proof. For the edge e we have two cases, $e \in Q$ or $e \in H$. If we have $e \in Q$ and $e \notin \mathcal{Q}$, then this edge will not be present in R, otherwise if $e \in \mathcal{Q}$ since P_1 covers all the edges of $\mathcal{Q}$ this edge will not be present in R. So the only case that $e \in R$ happens is when we have $e \in H$. In this case since we have $c_e \geq 1 - \tau$ and P_1 is a random minimum vertex cover of G, then the edge e will be covered with probability c_e in P_1 (this is because P_1 comes from the same distribution as OPT according to Claim 8). So this edge is present in P_2 with a probability of at most $1 - c_e \leq \tau$. ◁

The following claim will prove that the matching S_i are induced matchings.

▷ Claim 11. The matchings S_i formed in the random process are induced matchings.

Proof. The graphs R_i are induced subgraphs on $V - P_1$. Suppose that S_i's are not induced matchings. Since we delete duplicate edges from R_i to get S_i there is no edge e that is present in S_i and S_j for $i \neq j$. Hence, there must exist two edges $e_1 = (u_1, v_1)$ and $e_2 = (u_2, v_2)$ present in S_i and $e_3 = (u_1, u_2)$ present in S_j for $i \neq j$ to break the property of induced matchings. We call edges like e_3 a *cross edge*. In this case, we can argue that the edge e_3 is

also present in R_i since both of its endpoints are in the vertices of R_i. Since e_3 is present in R_i and R_j this edge gets deleted and can not be part of S_j. This contradicts with what we assumed. Therefore we can conclude that S_i's are induced matchings. ◁

▷ Claim 12. Suppose we set $T = \frac{1}{2\tau}$, then we have $\mathbf{E}[|S_i|] \geq \frac{k}{4}$.

Proof. Due to Lemma 9, we have that $\mathbf{E}[|M_i|] \geq \frac{k}{2}$. Now we should argue that with the edge deletions, each M_i will lose at most $\frac{k}{4}$ edges in expectation. Take an edge e in M_i. For each $j \neq i$ we have $\Pr[e \in M_i] \leq \tau$ due to Claim 10 since for an edge to be in M_i it should also be in R_i. By an application of union bound the edge e is present in at least one other M_j with probability at most $T\tau$. Hence we get,

$$\Pr[\text{e not in any } M_j] \geq 1 - T\tau = \frac{1}{2}$$

And for the size of S_i we have:

$$\mathbf{E}[|S_i|] = \sum_{e \in M_i} \mathbb{1}_{e \notin M_j \text{ for } j \neq i} \geq \sum_{e \in M_i} \frac{1}{2} \geq \frac{k}{4}$$ ◁

▶ **Lemma 13.** *By running the process explained above in Algorithm 2 we can find an $RS_n(r,t)$ graph (see Definition 2) with $t = \frac{1}{2\tau}$ and $r = \frac{k}{8}$ on H.*

Proof. Due to Claim 12 we can have $T = \frac{1}{2\tau}$ matchings each with expected size $\frac{k}{4}$. Putting these matchings together will give us expected $\frac{Tk}{4}$ edges in total. Suppose we break each matching S_i to smaller matchings of size $\frac{k}{8}$ size. If the number of edges in S_i is not divisible by $\frac{k}{8}$ we will lose at most the reminder of $|S_i|$ to $\frac{k}{8}$ which is at most $\frac{k}{8}$. The new matchings each have size exactly $\frac{k}{8}$ and are induced matchings. This is because the original S_i's did not have a cross edge (see Claim 11 for what a cross edge is) and hence the new matchings also do not have a cross edge since we are just partitioning the edges of S_i to smaller matchings. The total size of the new set of matchings is at least

$$\frac{Tk}{4} - \frac{Tk}{8} = \frac{Tk}{8}$$

This is because the total number of removed edges is bounded by the number of edges removed from each S_i which is $\frac{\varepsilon n}{8}$ times the number of S_i's which is T. Since we are running the random process which in expectation gives us induced matchings of size $\frac{k}{8}$ with total size of $\frac{Tk}{8}$ there is an instance which we have at least $\frac{Tk}{8}$ edges in total. From this we can construct an $RS_n(r,t)$ graph with $r = \frac{k}{8}$ and $t = \frac{1}{2\tau}$. ◀

Now we are ready to prove Theorem 5. We restate the theorem below.

▶ **Theorem 5.** *Let t be $RS_n(\varepsilon n/8)$. Running Algorithm 1 with $\tau = \frac{1}{3t}$ finds a vertex cover that has an additive approximation of at most εn to the optimal minimum vertex cover and queries a subgraph of size $O(tn/p)$.*

Proof. Suppose that Algorithm 1 does not find a vertex cover with additive approximation of εn. Then by setting $k = \varepsilon n$ in Lemma 13 we get

$$t' = \frac{1}{2\tau} = \frac{1}{2\frac{1}{3t}} = 1.5t$$

matchings each of size $\frac{\varepsilon n}{8}$ that are induced matchings. This is in contradiction with what we assumed since t is the maximum number of induced matchings of size $\frac{\varepsilon n}{8}$ for an RS-graph and now we have found $1.5t$ matchings of this size. This means that indeed Algorithm 1 has an additive approximation of at most εn for $\tau = \frac{1}{3t}$. Due to Claim 7, we have $|E(Q)| \leq O(\frac{n}{p\tau})$ which for $\tau = 1/3t$ we get $|E(Q)| = O(tn/p)$. ◀

3 Getting a Multiplicative Approximation

In this section, we modify Algorithm 1 to give us a multiplicative approximation of $1+\varepsilon$. Our modification is simple yet effective. At the beginning of the algorithm, we set $P_0 = \{v : c_v \geq 1 - \alpha\varepsilon\}$ and commit P_0 to the final solution. The intuition behind this is that since these vertices have high c_v, they have a high probability of being in the final solution so it does not hurt us to have them in the final solution.

We do this specifically to be able to prove Lemma 16 which states c_vs for a subgraph of vertices is lower bounded by $\frac{\alpha\varepsilon}{2}$. This means that if we find an algorithm that is additive in terms of n' the size of this subgraph, then it is also a multiplicative approximation because of the lower bound for c_vs of this subgraph. Here is the modified algorithm.

Algorithm 3 Multiplicative Approximation Algorithm.

1. Let $P_0 = \{v : c_v \geq 1 - \alpha\varepsilon\}$.
2. Let $G' = G[V \setminus P_0]$.
3. Let Q be the subgraph of edges with $c_e \leq 1 - \tau$ in G'.
4. Let $H = G' - Q$.
5. Query edges in Q to get its realization Q.
6. Hallucinate edges in $G \setminus Q$ to form subgraph $\mathcal{F}$.
7. Let $P_1 = \text{MVC}(Q \cup \mathcal{F})$.
8. Let $R = H[V \setminus P_1]$.
9. Let $P_2 = \text{MVC}(R)$.
10. Return $O = P_0 \cup P_1 \cup P_2$.

The first lemma we prove is for bounding the expected size of $P_0 \cup P_1$.

▶ **Lemma 14.** $\mathbf{E}[|P_0 \cup P_1|] \leq (1 + \varepsilon/2)\textsc{opt}$ *for* $\alpha \leq \frac{1}{4}$

Proof. We can write $E[|P_0 \cup P_1|]$ as follows:

$$E[|P_0 \cup P_1|] = |P_0| + \mathbf{E}[|P_1|] - \mathbf{E}[|P_0 \cap P_1|].$$

Due to Claim 8, we know that $\mathbf{E}[|P_1|] = \textsc{opt}$. Take an arbitrary vertex $v \in P_0$. Since P_1 is a random minimum vertex cover, any vertex v in P_0 has a probability of c_v of also being in P_1. And since $c_v \geq 1 - \alpha\varepsilon$ for $v \in P_0$ we have $\Pr[v \in P_1] \geq 1 - \alpha\varepsilon$. Hence we have

$$|P_0| - \mathbf{E}[|P_0 \cap P_1|] = \sum_{v \in P_0} 1 - \mathbb{1}_{v \in P_1} \leq |P_0| \cdot (\alpha\varepsilon) \tag{1}$$

We know that $|P_0| \leq \frac{\textsc{opt}}{1-\alpha\varepsilon}$ this is because $\textsc{opt} = \sum_{v \in V} c_v$ and we have $\sum_{v \in P_0} c_v \geq (1 - \alpha\varepsilon)|P_0|$. Form Equation (1), we can get

$$|P_0| - \mathbf{E}[|P_0 \cap P_1|] \leq \frac{\textsc{opt}}{1 - \alpha\varepsilon} \cdot (\alpha\varepsilon) \tag{2}$$

Since we have $\alpha \leq \frac{1}{4}$ and $\varepsilon \leq 1$ from Equation (2) we have

$$\begin{aligned} |P_0| - \mathbf{E}[|P_0 \cap P_1|] &\leq \frac{\textsc{opt}}{1 - \alpha\varepsilon} \cdot (\alpha\varepsilon) \\ &\leq \frac{\textsc{opt}}{1 - \alpha\varepsilon} \cdot (\frac{\varepsilon}{4}) \\ &\leq \frac{\textsc{opt}}{0.5} \cdot \frac{\varepsilon}{4} \\ &= \textsc{opt} \cdot \frac{\varepsilon}{2} \end{aligned}$$

Putting this together with $\mathbf{E}[|P_1|] = \textsc{opt}$ due to Claim 8 we get,

$$\mathbf{E}[|P_0 \cup P_1|] \leq (1 + \varepsilon/2)\textsc{opt}$$ ◀

From Lemma 14 we can argue that if Algorithm 3 has worse than εOPT additive approximation in expectation, then $|P_2|$ is greater than $\frac{\varepsilon\text{OPT}}{2}$. We use this fact in the following lemma.

▶ **Lemma 15.** *If Algorithm 3 does not have a approximation of at most $(1+\varepsilon)$ in expectation, then $\mathbf{E}[\mu(R)] \geq \frac{\varepsilon OPT}{4}$*

Proof. Due to Lemma 14 we have $\mathbf{E}[|P_0 \cup P_1|] \leq (1+\varepsilon/2)\text{OPT}$ so if Algorithm 3 does not have a multiplicative approximation of at most $(1+\varepsilon)$ then we can conclude $\mathbf{E}[|P_2|] \geq \frac{\varepsilon\text{OPT}}{2}$. In any graph, the maximum matching size is at least half the minimum vertex cover size. Applying this statement on the graph R gives us,

$$\mathbf{E}[\mu(R)] \geq \frac{E[|P_2|]}{2} \geq \frac{\varepsilon\text{OPT}}{4}$$ ◀

The next lemma proves a lower bound for c_v of vertices in H having at least one edge. We call this graph H for future reference, meaning we remove the vertices in H having no edge.

▶ **Lemma 16.** *For any vertex $v \in H$, we have $c_v \geq \frac{\alpha\varepsilon}{2}$ for $\tau \leq \frac{\alpha\varepsilon}{2}$.*

Proof. Take an edge $e = (u, v) \in H$ since it has a degree of at least one. For this edge we know that $c_e \geq 1 - \tau$ since we put all the edges with $c_e \leq 1 - \tau$ in Q. Also we have $c_u < 1 - \alpha\varepsilon$ since $u \notin P_0$. We also know that $c_e \leq c_v + c_u$. Putting this all together we get that $c_v \geq \alpha\varepsilon - \tau$ which for a $\tau \leq \frac{\alpha\varepsilon}{2}$ we get $c_v \geq \frac{\alpha\varepsilon}{2}$. ◀

▷ Claim 17. We have $\text{OPT} \geq \frac{n'\alpha\varepsilon}{2}$ for $n' = |V(H)|$

Proof. This is because we have,

$$\text{OPT} = \sum_{v \in V} c_v \geq \sum_{v \in H'} c_v$$

and we know that $\sum_{v \in H'} c_v \geq n' \cdot \frac{\alpha\varepsilon}{2}$ due to Lemma 16. ◁

Putting Claim 17 and Lemma 15 together we get that if Algorithm 3 does not give a $(1+\varepsilon)$ approximation, then $\mathbf{E}[\mu(R)] \geq \frac{n'\alpha\varepsilon^2}{8}$. The next lemma asserts that if Algorithm 3 does not find a $(1+\varepsilon)$ approximation then we can find a large RS-graph in H.

▶ **Lemma 18.** *If Algorithm 3 is not a $(1+\varepsilon)$ approximation then, we can find an $RS_{n'}(r,t)$ graph (see Definition 2) with $t = 1/2\tau$ and $r = \frac{\alpha\varepsilon^2 n'}{32}$ on H*

Proof. Suppose we set $k = \frac{n'\alpha\varepsilon^2}{4}$. Then we can run the algorithm explained in Algorithm 2 and put together matchings S_i formed from different realizations of H to find an RS-graph where each matching size is $k/8$. Due to having $\mathbf{E}[\mu(R)] \geq \frac{n'\alpha\varepsilon^2}{8}$ we can prove Claim 10. This is because for edges in R_i we can argue that they exist in R_i with a probability of at most τ similar to how we proved it for R_i's in Algorithm 1 in Claim 10. Moreover, like how we proved for Algorithm 2 that the matchings S_i are induced matchings we form S_i's like in Algorithm 2 and we can argue that they are induced matchings. Then we can also prove Claim 12 and Lemma 13 for the new value of k. Therefore, similar to how we proved Lemma 13 we can find an $\text{RS}_{n'}(r,t)$ graph with $t = \frac{1}{2\tau}$ and $r = \frac{k}{8} = \frac{\alpha\varepsilon^2 n'}{32}$ on H. ◀

Now we are ready to prove the main result of this work.

▶ **Result 3.** *Given any $\varepsilon \in (0,1)$ let t be $RS_n(cn)$ for $c = \frac{\varepsilon^2}{128}$. There exists an algorithm that queries a subgraph of size $O(tn/p)$ and finds a vertex cover with the expected approximation ratio of $(1+\varepsilon)$.*

Proof. Let us run Algorithm 3 with $\alpha = 1/4$ and $\tau = 1/3t$. Meaning we have $c = \frac{\alpha\varepsilon^2}{32}$. We proceed similar to how we proved Theorem 5. Suppose that Algorithm 3 does not find a vertex cover with $(1+\varepsilon)$ approximation. Then we can find an $RS_{n'}(r,t)$ graph in H for $t = 1/2\tau$ and $r = \frac{\alpha\varepsilon^2 n'}{32}$ since $\mathrm{RS}_n(cn) \geq \mathrm{RS}_{n'}(cn')$ for $n \geq n'$. By setting $k = \frac{n'\alpha\varepsilon^2}{4}$ in Lemma 13 we get

$$t' = \frac{1}{2\tau} = \frac{1}{2\frac{1}{3t}} = 1.5t$$

matchings each of size $\frac{k}{8}$ that are induced matchings. This is in contradiction with what we assumed since t is the maximum number of induced matchings of size $\frac{k}{8}$ for an RS-graph and now we have found $1.5t$ matchings of this size. This means that indeed Algorithm 3 has an approximation of at most $(1+\varepsilon)$ for $\tau = \frac{1}{3t}$. Due to Claim 7, we have $|E(Q)| \leq O(\frac{n}{p\tau})$ which for $\tau = 1/3t$ we get $|E(Q)| = O(tn/p)$. ◀

Correlated Edges

Suppose that G has some edges that are realized in $\mathcal{G}$ independently called G_1 and some edges that are correlated with each other and the edges in G_1 called G_2. This means that edges in G_2 might exist based on edges in G_1 or G_2 that are realized. We need to adjust the definition of hallucination and realize the correlated edges by drawing them from the given distribution rather than realizing them independently. Moreover, if a set of them are queried and we want to hallucinate the rest, it should be done conditionally.

Notice that the only place we use independent realization of edges is in the proof of Claim 7. So we can argue that among the edges in G_1 at most $O(tn/p)$ edges will be in Q. Furthermore we know that $|Q| \leq |Q_1| + |Q_2|$ where Q_i is the subgraph of G_i that is in Q. Since the number of correlated edges is at most $\mathrm{RS}_n(cn)$ we have $|Q_2| \leq n \cdot \mathrm{RS}_n(cn)$ so we get:

$$|Q| \leq |Q_1| + |Q_2| \leq O(tn/p) + n \cdot \mathrm{RS}_n(cn) = O(tn/p) + n \cdot t = O(tn/p)$$

Based on the explanation above we drive the following corollary.

▶ **Corollary 19.** *Our Algorithm 3 can handle up to $t = RS_n(cn)$ correlated edges for $c = \frac{\varepsilon^2}{128}$ and still provide a $(1+\varepsilon)$ approximation to the optimal solution with $O(tn/p)$ queries.*

References

1 Sepehr Assadi. A two-pass (conditional) lower bound for semi-streaming maximum matching. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 708–742. SIAM, 2022. doi:10.1137/1.9781611977073.32.

2 Sepehr Assadi, Soheil Behnezhad, Sanjeev Khanna, and Huan Li. On regularity lemma and barriers in streaming and dynamic matching. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 131–144. ACM, 2023. doi:10.1145/3564246.3585110.

3 Sepehr Assadi and Aaron Bernstein. Towards a unified theory of sparsification for matching problems. In Jeremy T. Fineman and Michael Mitzenmacher, editors, *2nd Symposium on Simplicity in Algorithms, SOSA 2019, January 8-9, 2019, San Diego, CA, USA*, volume 69 of *OASIcs*, pages 11:1–11:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/OASICS.SOSA.2019.11.

4 Sepehr Assadi and Sanjeev Khanna. Improved bounds for fully dynamic matching via ordered ruzsa-szemerédi graphs. *CoRR*, abs/2406.13573, 2024. doi:10.48550/arXiv.2406.13573.

5 Sepehr Assadi, Sanjeev Khanna, and Yang Li. The stochastic matching problem: Beating half with a non-adaptive algorithm. In Constantinos Daskalakis, Moshe Babaioff, and Hervé Moulin, editors, *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, pages 99–116. ACM, 2017. doi:10.1145/3033274.3085146.

6 Sepehr Assadi, Sanjeev Khanna, and Yang Li. The stochastic matching problem with (very) few queries. *ACM Trans. Economics and Comput.*, 7(3):16:1–16:19, 2019. doi:10.1145/3355903.

7 Soheil Behnezhad, Avrim Blum, and Mahsa Derakhshan. Stochastic vertex cover with few queries. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 1808–1846. SIAM, 2022. doi:10.1137/1.9781611977073.73.

8 Soheil Behnezhad and Mahsa Derakhshan. Stochastic weighted matching: (stochastic weighted matching: (1-ε) approximation. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1392–1403. IEEE, 2020. doi:10.1109/FOCS46700.2020.00131.

9 Soheil Behnezhad, Mahsa Derakhshan, Alireza Farhadi, MohammadTaghi Hajiaghayi, and Nima Reyhani. Stochastic matching on uniformly sparse graphs. In Dimitris Fotakis and Evangelos Markakis, editors, *Algorithmic Game Theory - 12th International Symposium, SAGT 2019, Athens, Greece, September 30 - October 3, 2019, Proceedings*, volume 11801 of *Lecture Notes in Computer Science*, pages 357–373. Springer, 2019. doi:10.1007/978-3-030-30473-7_24.

10 Soheil Behnezhad, Mahsa Derakhshan, and MohammadTaghi Hajiaghayi. Stochastic matching with few queries: (1-ε) approximation. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1111–1124. ACM, 2020. doi:10.1145/3357713.3384340.

11 Soheil Behnezhad and Alma Ghafari. Fully dynamic matching and ordered ruzsa-szemerédi graphs. In *65st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024*, 2024.

12 Soheil Behnezhad and Nima Reyhani. Almost optimal stochastic weighted matching with few queries. In Éva Tardos, Edith Elkind, and Rakesh Vohra, editors, *Proceedings of the 2018 ACM Conference on Economics and Computation, Ithaca, NY, USA, June 18-22, 2018*, pages 235–249. ACM, 2018. doi:10.1145/3219166.3219226.

13 Avrim Blum, John P. Dickerson, Nika Haghtalab, Ariel D. Procaccia, Tuomas Sandholm, and Ankit Sharma. Ignorance is almost bliss: Near-optimal stochastic matching with few queries. In Tim Roughgarden, Michal Feldman, and Michael Schwarz, editors, *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*, pages 325–342. ACM, 2015. doi:10.1145/2764468.2764479.

14 Mahsa Derakhshan, Naveen Durvasula, and Nika Haghtalab. Stochastic minimum vertex cover in general graphs: A 3/2-approximation. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 242–253. ACM, 2023. doi:10.1145/3564246.3585230.

15 Mahsa Derakhshan and Mohammad Saneian. Query efficient weighted stochastic matching. *CoRR*, abs/2311.08513, 2023. doi:10.48550/arXiv.2311.08513.

16 Shaddin Dughmi, Yusuf Hakan Kalayci, and Neel Patel. On sparsification of stochastic packing problems. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*, volume 261 of *LIPIcs*, pages 51:1–51:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.ICALP.2023.51.

17 Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 474–483, 2002. doi:10.1145/509907.509977.

18 Jacob Fox. A new proof of the graph removal lemma. *Annals of Mathematics*, pages 561–579, 2011.

19 Jacob Fox, Hao Huang, and Benny Sudakov. On graphs decomposable into induced matchings of linear sizes. *Bulletin of the London Mathematical Society*, 49(1):45–57, 2017.

20 Ashish Goel, Michael Kapralov, and Sanjeev Khanna. On the communication and streaming complexity of maximum bipartite matching. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 468–485. SIAM, 2012. doi:10.1137/1.9781611973099.41.

21 Michel X. Goemans and Jan Vondrák. Covering minimum spanning trees of random subgraphs. In J. Ian Munro, editor, *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, New Orleans, Louisiana, USA, January 11-14, 2004*, pages 934–941. SIAM, 2004. URL: http://dl.acm.org/citation.cfm?id=982792.982933.

22 Michel X. Goemans and Jan Vondrák. Covering minimum spanning trees of random subgraphs. *Random Struct. Algorithms*, 29(3):257–276, 2006. doi:10.1002/RSA.20115.

23 Imre Z Ruzsa and Endre Szemerédi. Triple systems with no six points carrying three triangles. *Combinatorics (Keszthely, 1976), Coll. Math. Soc. J. Bolyai*, 18(939-945):2, 1978.

24 Jan Vondrák. Shortest-path metric approximation for random subgraphs. *Random Struct. Algorithms*, 30(1-2):95–104, 2007. doi:10.1002/RSA.20150.

25 Yutaro Yamaguchi and Takanori Maehara. Stochastic packing integer programs with few queries. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 293–310. SIAM, 2018. doi:10.1137/1.9781611975031.21.

Nearest Neighbor Complexity and Boolean Circuits

Mason DiCicco
Worcester Polytechnic Institute, MA, USA

Vladimir Podolskii
Tufts University, Medford, MA, USA

Daniel Reichman
Worcester Polytechnic Institute, MA, USA

Abstract

A nearest neighbor representation of a Boolean function f is a set of vectors (anchors) labeled by 0 or 1 such that $f(\boldsymbol{x}) = 1$ if and only if the closest anchor to $\boldsymbol{x}$ is labeled by 1. This model was introduced by Hajnal, Liu and Turán [2022], who studied bounds on the minimum number of anchors required to represent Boolean functions under different choices of anchors (real vs. Boolean vectors) as well as the analogous model of k-nearest neighbors representations.

We initiate a systematic study of the representational power of nearest and k-nearest neighbors through Boolean circuit complexity. To this end, we establish a close connection between Boolean functions with polynomial nearest neighbor complexity and those that can be efficiently represented by classes based on linear inequalities – *min-plus polynomial threshold functions* – previously studied in relation to threshold circuits. This extends an observation of Hajnal et al. [2022]. Next, we further extend the connection between nearest neighbor representations and circuits to the k-nearest neighbors case.

As an outcome of these connections we obtain exponential lower bounds on the k-nearest neighbors complexity of explicit n-variate functions, assuming $k \leq n^{1-\epsilon}$. Previously, no superlinear lower bound was known for any $k > 1$. At the same time, we show that proving superpolynomial lower bounds for the k-nearest neighbors complexity of an explicit function for arbitrary k would require a breakthrough in circuit complexity. In addition, we prove an exponential separation between the nearest neighbor and k-nearest neighbors complexity (for unrestricted k) of an explicit function. These results address questions raised by [17] of proving strong lower bounds for k-nearest neighbors and understanding the role of the parameter k. Finally, we devise new bounds on the nearest neighbor complexity for several families of Boolean functions.

2012 ACM Subject Classification Theory of computation → Complexity classes

Keywords and phrases Complexity, Nearest Neighbors, Circuits

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.42

Related Version *Previous Version*: `https://arxiv.org/pdf/2402.06740`

Acknowledgements We would like to thank Bill Martin for several insightful comments. The second and third author thank the Simons Institute for the Theory of Computing where collaboration on this project has began.

1 Introduction

A *nearest-neighbor representation* of a function $f : \{0,1\}^n \to \{0,1\}$ is a set of vectors, called "anchors," say $S = P \cup N$ such that $f(\boldsymbol{x}) = 1$ if and only if the nearest anchor to $\boldsymbol{x}$ (under the Euclidean distance metric) belongs to P. The set of anchors can be seen as a (disjoint) union of "positive" and "negative" examples. If $S \subseteq \{0,1\}^n$, we refer to the representation as *Boolean*, and if $S \subseteq \mathbb{R}^n$ we call it *real*. This model was pioneered by [17], who advocated the study of Boolean functions admitting efficient representations, focusing on the *number of anchors* as a measure of the complexity of the representation. They also consider the *k-nearest neighbors* variation, where the value of f on input x is computed as a majority vote of the k nearest anchors to x.


© Mason DiCicco, Vladimir Podolskii, and Daniel Reichman;
licensed under Creative Commons License CC-BY 4.0



In particular, [17] observed a relationship between nearest-neighbor representations and threshold circuits. Motivated by their work, we initiate a systematic study of the connections to circuit complexity. Some of our results are related to the *weight complexity* (number of bits) needed when representing Boolean functions by real anchors: A topic receiving recent attention by [24].

There are numerous extensions and variations of nearest neighbor complexity. While studying all of these here is infeasible, one goal of ours is to encourage further exploration of this broad topic. We discuss some future directions in the conclusion section.

1.1 Motivation

We believe that nearest neighbor representations are a natural and interesting model of computation worthy of study. Furthermore, nearest neighbor complexity is relevant to several central research topics listed next.

1.1.1 Boolean function complexity

Nearest neighbor representations are related to depth-two threshold circuits and decision trees: [17] establish that lower bounds for these models are useful for establishing lower bounds for nearest neighbors. Polynomial threshold functions and their relation to geometric representations of Boolean functions also bear a strong connection to nearest neighbor representations. In fact, we will show that the expressivity of nearest neighbors is essentially equivalent to that of polynomial threshold functions *over the tropical semiring* – An interesting model of computation due to [18].

Our approach to understanding nearest neighbor complexity in terms of Boolean circuit complexity follows a long tradition in computational learning theory and computational complexity (e.g., [29, 22, 26, 27, 41, 39]). Uncovering new connections between nearest neighbors and well-studied computational models (such as linear decision lists and depth-two circuits) enables us to prove new unconditional lower bounds and upper bounds on the nearest neighbor complexity of explicit[1] functions. It also allows us to phrase some open problems in circuit complexity in terms of nearest neighbor complexity. For instance, the difficulty of proving super-polynomial lower bounds for k-nearest neighbors representations – for appropriate values of k – follows from representations (that we construct) of circuits with the same difficulty.

1.1.2 Machine learning and pattern recognition

Learning algorithms based on nearest neighbors have been the subject of extensive research for more than 50 years [8, 11]. For example, there is evidence that increasing k in the k-nearest neighbors rule can decrease its estimation error [10]. However, much less seems to be known about the *capacity* of the nearest and k-nearest neighbor rules to represent certain functions, while the capacity of other machine learning models such as Boolean circuits and neural networks has received considerable interest [16, 19, 30, 12, 38].

[1] By "explicit," we mean a function that can be computed in polynomial time by a Turing machine.

1.1.3 Algorithms for nearest neighbors classification and search

Efficient implementation of the nearest neighbor rule in high dimension has received considerable interest, leading to efficient algorithms and sophisticated data structures [7, 2, 1]. The study of nearest neighbor complexity leads naturally to the study of nonuniform algorithms (Boolean circuits) for this rule. Our work (along with previous findings) yields upper and lower bounds on the size of a circuit needed to implement nearest neighbor classification. We focus on exact representations, leaving the study of circuit complexity for *approximate* nearest neighbor search (e.g., [20, 28]) for future work.

1.2 Our results

The objects of our study are the *classes* of Boolean functions admitting *real* and *Boolean* nearest neighbor representation of polynomial complexity, NN and HNN respectively. Standard complexity classes based on circuit-like models of computation are closed under two natural "rewiring" operations: Substitution of variables by constants and duplication of variables (See Definition 11). However, it is not clear how to efficiently perform these operations in the nearest neighbor model. Thus, if we would like to give a precise characterization of NN and HNN in terms of circuits, we have to consider the *closure* of these classes under the same operations. As a result, we obtain classes of *subfunctions* of polynomial-size nearest neighbor representations, $\overline{\mathsf{NN}}$ and $\overline{\mathsf{HNN}}$ (see Defintion 12). We then give a precise characterization (equality) of $\overline{\mathsf{NN}}$ and $\overline{\mathsf{HNN}}$ in terms of the class of *min-plus polynomial threshold functions* of polynomial complexity (mpPTF)[2]. This adds to previous results of [17] showing the *containment* of NN and HNN in mpPTF. As a consequence, we prove (among other results) that HNN contains functions that cannot be computed by depth-two threshold circuits with polynomial size and weight. We also observe that the closure of HNN is closely connected to the class of functions with NN representations of logarithmic bit-complexity.

We study the k-nearest neighbors complexity of Boolean functions for $k > 1$. First, we extend the aforementioned characterization – the closure of NN in terms of mpPTF – to the closure of kNN. We use this characterization to prove that kNN for *constant* k is closely related to NN and that there exists an explicit function that requires exponential kNN complexity when $k \leq n^{1-\epsilon}$ (for an n-variate function). Next, we generalize the characterization of kNN to *arbitrary* k by introducing a new class, kSTAT – functions realizable by an inequality of the k-statistics of two sets of linear forms – which generalizes mpPTF. Consequently, we show that proving lower bounds for kNN for *arbitrary* k would result in lower bounds for the circuit class $\mathsf{SYM} \circ \mathsf{MAJ}$, which would be a major breakthrough in Boolean circuit complexity.

Finally, we present new bounds for nearest neighbor complexity of specific Boolean functions such as disjointness, CNFs, and majority. For example, we show that CNFs with polynomially many clauses have NN representations with polynomially-many anchors which also exhibit *constant bit-complexity*. In contrast, there exist CNFs of polynomial size with exponential HNN complexity. We also establish a new lower bound of $n/2 + 2$ on the HNN complexity of the majority function with an even number of inputs (n). This lower bound is tight, as it matches the upper bound proved in [17].

[2] An mpPTF is an expression of the form $\min\{L_1(\boldsymbol{x}), \cdots, L_\ell(\boldsymbol{x})\} \leq \min\{R_1(\boldsymbol{x}), \cdots, R_r(\boldsymbol{x})\}$ where L_i, R_i are linear forms.

1.3 Related work

Nearest neighbor complexity (under Euclidean distance) was formalized by [17]. They prove that the functions[3] $\mathsf{THR}^{\lfloor n/3 \rfloor}$ and XOR both require an exponential number of Boolean anchors but only 2 and $n+1$ real anchors, respectively. In fact, the same argument proves that THR^t requires at least $\binom{n}{t}/\binom{2t}{t}$ anchors for any t, which gives exponential lower bound for t bounded away from $n/2$, but is vacuous for $\mathsf{THR}^{n/2}$. It was subsequently shown in [24] that any symmetric Boolean function f has an NN representation with $I(f)$ anchors, where $I(f)$ denotes the number of *intervals* of f – the minimal number of contiguous intervals $[a, b]$ partitioning $[0, n]$ where $f(\boldsymbol{x})$ is constant for $a \leq \sum_i x_i \leq b$ – and this bound is optimal when all intervals have the same length. This extends the result of [17] that every symmetric function has nearest neighbor complexity at most $n+1$.

1.3.1 Connections to circuits

It was observed in [17] that functions with polynomial nearest neighbor complexity can be simulated by min-plus polynomial threshold functions, but it is an open question of whether or not the inclusion $\mathsf{NN} \subseteq \mathsf{mpPTF}$ is proper. Relations to the class mpPTF are of interest because it deepens the connection between nearest neighbors and circuit complexity. For instance, [18] establish that systems of mpPTFs compute exactly the class of $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{THR}$ circuits.

The expressive power of k-nearest neighbors rule was also studied by [17]. In particular, they prove that kNN can simulate *linear decision trees*, which yields a linear (in n) lower bound for the number of anchors in kNN representations of the $\mathsf{IP}\mod 2$ function. They also state the open problem of proving stronger lower bounds for the k-nearest neighbors complexity of an explicit function. In [25], it is shown that, under some regularity assumptions, polynomial-size nearest neighbor representations can simulate convex polytopes (i.e., $\mathsf{AND} \circ \mathsf{THR}$) as well as logarithmic fan-in $\mathsf{SYM} \circ \mathsf{THR}$ circuits and linear decision lists (LDL).

Constructions of Boolean circuits computing nearest neighbor classification are known. [31] constructs an $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{THR}$ circuit computing any function with an m-anchor NN representation in size $O(m^2)$. (See Appendix B). A very similar depth-three construction for kNN, also with size $O(m^2)$, was found by [6]. Note that the weights of the above circuits are bounded by a polynomial in n.

1.3.2 Bit complexity

It was shown in [24] that (the aforementioned) NN representations for symmetric functions have *logarithmic bit-complexity*, and that this is tight for some functions. It is left as an open problem to characterize NN representations of *threshold functions* in terms of bit-complexity. To this end, the same authors (in [25]) show that logarithmic bit complexity suffices to represent the comparison, equality, and odd-max bit Boolean functions, and conjecture that a logarithmic upper bound holds for any threshold function.

Other works have studied the role of bit-complexity in *approximate nearest neighbor search*; where we wish to find an anchor whose distance is minimal to a query point, up to a factor of $(1+\epsilon)$. For example, [21] provide tight bounds (in terms of bit-complexity) on the size of data structures performing approximate nearest neighbor search. This setting is quite different from our focus on exact classification of Boolean vectors.

---

[3] $\mathsf{THR}^t(\boldsymbol{x}) = 1 \iff \sum_i x_i \geq t$

The bit-complexity of the weights in polynomial-size threshold circuits has been studied extensively (see, e.g., surveys [34, 37]). For example, it was proved by [14, 15] that arbitrarily large weights can be reduced to have logarithmic bit-complexity by slightly increasing the depth of the circuit (along with a polynomial blow-up in size).

1.4 Organization

Section 2 outlines basic definitions required in subsequent sections. Section 3 establishes the equivalence between $\overline{\mathsf{HNN}}, \overline{\mathsf{NN}}$ and min-plus polynomial threshold functions, then discusses some of the consequences. Section 4 generalizes mpPTF to a new class, kSTAT, and proves that a similar equivalence holds with $\overline{\mathsf{kNN}}$. Here, we also derive several connections to circuit classes such as $\mathsf{SYM} \circ \mathsf{MAJ}$. Section 5 contains new results (upper and lower bounds) for the nearest neighbor complexity of explicit Boolean functions. Many proofs are relegated to Appendix A due to space constraints. Appendix B contains some direct constructions of threshold circuits computing HNN, one of which has depth two.

2 Preliminaries

We use the following notation throughout the paper:

Vectors are written in bold (i.e. $\boldsymbol{x} = (x_1, \cdots, x_n)$).
The k'th statistic of $\boldsymbol{x}$, denoted $\boldsymbol{x}_{(k)}$, is the k'th smallest element of $\boldsymbol{x}$.
– In particular, $\boldsymbol{x}_{(k)} = x_{\sigma(k)} \iff x_{\sigma(1)} \leq \cdots \leq x_{\sigma(n)}$ for some $\sigma \in S_n$
$\Delta(\boldsymbol{x}, \boldsymbol{y}) := ||\boldsymbol{x} - \boldsymbol{y}||_2^2$ denotes the squared Euclidean distance between $\boldsymbol{x}, \boldsymbol{y}$.
$\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ denotes the real inner product (dot product), $x_1 y_1 + \cdots + x_n y_n$.
$\mathrm{poly}(n)$ refers to an arbitrary polynomial in the variable n.
$\mathbb{1}[P]$ denotes the Boolean function whose value is 1 if and only if P holds.

Note that the (squared) Euclidean distance between two Boolean vectors is equal to their Hamming distance, $\Delta(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i \leq n} \mathbb{1}[x_i \neq y_i]$, so the Hamming weight of a Boolean vector $\boldsymbol{p}$ is denoted $\Delta(\boldsymbol{p}) := \Delta(\boldsymbol{p}, \boldsymbol{0}) = ||\boldsymbol{p}||_2^2$.

2.1 Boolean functions

▶ **Definition 1.** *A* threshold gate *is a Boolean function* $f : \{0,1\}^n \to \{0,1\}$ *defined by a weight vector* $\boldsymbol{w} \in \mathbb{R}^n$ *and a threshold* $\theta \in \mathbb{R}$ *such that*

$$f(\boldsymbol{x}) = 1 \iff \langle \boldsymbol{w}, \boldsymbol{x} \rangle \geq \theta. \tag{1}$$

A threshold circuit *is a sequence* $(f_1, \cdots, f_s)$ *of* $s \geq n$ *gates such that the first* n *gates are equal to the input variables (i.e.,* $f_i = x_i$ *for* $i \leq n$*) and subsequent gates are threshold gates whose inputs are some subset of the previous gates. The output of the circuit is equal to the output of the final gate. The size of the circuit is equal to* $s - n$.

A threshold circuit can be viewed as a directed acyclic graph. Nodes with fan-in 0 correspond to inputs, and other nodes correspond to threshold gates applied to the values of the preceding nodes. The node with fan-out 0 correspond to the output node. The *depth* of the circuit is the length of the longest path from an input node to the output node.

▶ Remark 2. It is well known that we may assume that the weights (and the threshold) are integers without loss of generality: Since the domain of a threshold gate is finite, we may approximate each weight by a rational number and multiply by a common denominator. See [23] for a comprehensive introduction to circuit complexity.

▶ **Definition 3.** *A* Nearest Neighbor (NN) *representation of a Boolean function* $f : \{0,1\}^n \to \{0,1\}$ *is defined by two disjoint sets of positive and negative* anchors $P, N \subseteq \mathbb{R}^n$ *such that*

- $f(\boldsymbol{x}) = 1$ *if there exists a* $\boldsymbol{p} \in P$ *with* $\Delta(\boldsymbol{x}, \boldsymbol{p}) < \Delta(\boldsymbol{x}, \boldsymbol{q})$ *for all* $\boldsymbol{q} \in N$.
- $f(\boldsymbol{x}) = 0$ *if there exists a* $\boldsymbol{q} \in N$ *with* $\Delta(\boldsymbol{x}, \boldsymbol{q}) < \Delta(\boldsymbol{x}, \boldsymbol{p})$ *for all* $\boldsymbol{p} \in P$.

A Hamming Nearest Neighbor (HNN) *representation is defined identically for Boolean anchors in* $\{0,1\}^n$.

▶ **Definition 4.** *A* k-Nearest Neighbors (kNN) *representation of a function* $f : \{0,1\}^n \to \{0,1\}$ *is defined by two disjoint sets of positive and negative* anchors $P, N \subseteq \mathbb{R}^n$ *and an integer* k *such that*

$$f(\boldsymbol{x}) = 1 \iff \text{there exists an } A \subseteq P \cup N \text{ with the following properties:}$$

1. $|A| = k$
2. $|A \cap P| \geq |A \cap N|$
3. $\Delta(\boldsymbol{x}, \boldsymbol{a}) < \Delta(\boldsymbol{x}, \boldsymbol{b})$ *for all* $\boldsymbol{a} \in A$, $\boldsymbol{b} \notin A$.

A kHNN *representation is defined identically for Boolean anchors.*

▶ **Definition 5** ([18])**.** *A* min-plus polynomial threshold function (mpPTF) *is a Boolean function* $f : \{0,1\}^n \to \{0,1\}$ *defined by two sets of linear forms with integer coefficients*[4] $\{L_1, \cdots, L_{\ell_1}\} \cup \{R_1, \cdots, R_{\ell_2}\}$ *satisfying*

$$f(\boldsymbol{x}) = 1 \iff \min_{i \leq \ell_1} L_i(\boldsymbol{x}) \leq \min_{j \leq \ell_2} R_j(\boldsymbol{x}) \tag{2}$$

The number of terms *in an* mpPTF *is equal to* $\ell_1 + \ell_2$*, and the* maximum weight *is equal to the largest absolute value of the coefficients of any form.*

▶ **Definition 6** ([36])**.** *A* linear decision list (LDL) *representation of a Boolean function* f *is a sequence of instructions "if* $f_i(\boldsymbol{x}) = 1$*, then output* c_i *(and halt)" for* $1 \leq i \leq m$*, followed by "output 0." Here,* $f_1, \cdots, f_m$ *are threshold gates and* $c_1, \cdots, c_m \in \{0,1\}$. Exact *linear decision lists (*ELDL*) are defined similarly using* exact *threshold functions – threshold gates where the inequality in* (1) *is replaced with equality. The* length *of an* LDL *or* ELDL *is the number of gates,* m*, and its* maximum weight *is equal to the largest coefficient of any* f_i.

▶ **Definition 7.** *We consider the following well-known Boolean functions.*

The majority *function,* $\mathsf{MAJ}(x_1, \cdots, x_n) = \mathbb{1}[x_1 + \cdots + x_n \geq n/2]$
The disjointness *function,* $\mathsf{DISJ}(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{1}[\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0]$
The inner product mod 2 *function,* $\mathsf{IP}(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{y} \rangle \mod 2$
The odd-max-bit *function,* $\mathsf{OMB}(x_1, \cdots, x_n) = \max\{i : x_i = 1\} \mod 2$

2.2 Function classes

First, we define classes of Boolean circuits whose inputs may be variables, their negations, or the constants 0 and 1. AND, OR, THR, and SYM are the classes of polynomial-size[5] depth-one circuits composed of AND, OR, threshold gates, and symmetric functions (i.e., Boolean functions which depend only on the Hamming weight of the input) respectively. $\mathsf{MAJ} \subset \mathsf{THR}$ is the set of threshold gates with polynomial weights[6]. AC^0 is the class of constant-depth circuits consisting of a polynomial number of AND, OR, and NOT gates.

[4] As for threshold gates, there is no loss of generality in the assumption that weights of mpPTFs are integers.
[5] "polynomial" in this context is always with respect to the input size, n.
[6] We abuse the notation denoting by MAJ both specific function and a class of function. The meaning of our notation will be also clear from the context.

For two circuit classes C_1, C_2, the class of circuits consisting of a circuit from C_1 whose inputs are (the outputs of) a polynomial number of circuits from C_2 is denoted by $C_1 \circ C_2$. (e.g., $\mathsf{THR} \circ \mathsf{THR}$ refers to depth two threshold circuits of polynomial size.)

▶ **Definition 8.** NN *is the class of Boolean functions that have nearest neighbor representations with polynomially-many anchors.* HNN *is the same class where anchors are Boolean.* kNN *and* kHNN *are defined in the same manner for a positive integer* k*.*

▶ **Definition 9.** $\mathsf{mpPTF}(\infty)$ *is the class of min-plus polynomial threshold functions with a polynomial number (in terms of the number of inputs) of terms and unbounded maximum weight.* $\mathsf{mpPTF}(\mathrm{poly}(n))$ *is the same class with polynomially-bounded maximum weight.*

▶ **Definition 10.** LDL *is the class of Boolean functions representable by linear decision lists with polynomial length.* $\widehat{\mathsf{LDL}}$ *is the same class with polynomially-bounded maximum weight.* ELDL *and* $\widehat{\mathsf{ELDL}}$ *are defined similarly for exact linear decision lists.*

3 Min-plus PTFs vs. nearest neighbors

In this section, we introduce the *closure* operation and derive an equivalence between (the closure of) NN, HNN and mpPTF.

▶ **Definition 11.** *Define a substitution of variables as a function* $v : \{0,1\}^n \rightsquigarrow \{0,1\}^{\widetilde{n}}$ *where* $\rightsquigarrow$ duplicates *variables or* adds constant *variables (e.g.,* $x_1x_2 \rightsquigarrow x_1x_1x_2x_2x_20$*). Then, a Boolean function* $f : \{0,1\}^n \to \{0,1\}$ *is a* subfunction *of* $g : \{0,1\}^{\widetilde{n}} \to \{0,1\}$ *when* $\widetilde{n} = \mathrm{poly}(n)$ *and there exists a substitution of variables* v *such that* $f(\boldsymbol{x}) = g(v(\boldsymbol{x}))$ *for all* $\boldsymbol{x} \in \{0,1\}^n$*.*

Subfunctions may equivalently be obtained from $g : \{0,1\}^{\widetilde{n}} \to \{0,1\}$ by *identifying variables* (e.g., $x_1 = x_2$) and *assigning variables* to constants (e.g., $x_1 = 0$).

▶ **Definition 12.** *For any function class* C*, let* $\overline{C}$ *denote the* closure *of* C*: The set of subfunctions derived from the elements of* C*. In particular, we say that a Boolean function* f *has an "*$\overline{\mathsf{NN}}$ *representation" if it is a subfunction of some* $g \in \mathsf{NN}$*.*

▶ Note 13. Circuit classes are already closed under this operation. For example, $\overline{\mathsf{MAJ}} = \mathsf{MAJ}$: Subfunctions of the majority function simply add (polynomially-bounded) coefficients and constant terms.

▶ **Theorem 14.**

$$\overline{\mathsf{NN}} = \mathsf{mpPTF}(\infty), \quad \overline{\mathsf{HNN}} = \mathsf{mpPTF}(\mathrm{poly}(n))$$

Theorem 14 and some consequences are proved in Appendix A.1. Namely, we observe that any n-variate function in $\overline{\mathsf{NN}}$ is a sub-function of an $(n+1)$-variate NN representation, and that $\mathsf{mpPTF}(\mathrm{poly}(n))$ captures precisely the power of $\overline{\mathsf{NN}}$ representations with *bit-complexity* $O(\log n)$. Then, using the results of [17] and [18], we immediately establish the following two corollaries.

▶ **Corollary 15.** $\mathsf{HNN} \subsetneq \overline{\mathsf{HNN}}$

▶ **Corollary 16.** $\overline{\mathsf{NN}}$ *representations of* IP *and* $f_n(\boldsymbol{x}, \boldsymbol{y}) := \bigwedge_{i=1}^{n} \bigvee_{j=1}^{n^2} (x_{i,j} \wedge y_{i,j})$ *require* $2^{\Omega(n)}$ *anchors.*

(Proofs in A.2 and A.3.) Theorem 14 also yields lower bounds for the circuit complexity of functions belonging to HNN. (A direct construction in Appendix B shows that $\mathsf{HNN} \subseteq \mathsf{THR} \circ \mathsf{MAJ}$.)

▶ **Theorem 17.**

$$\mathsf{HNN} \not\subseteq \mathsf{MAJ} \circ \mathsf{MAJ}$$

More precisely, there is a Boolean function with an HNN *representation with* $n+1$ *anchors which cannot be computed by a depth-two majority circuit with* $\mathrm{poly}(n)$ *gates.*

Proof. First, we claim that $\mathsf{OMB} \circ \mathsf{AND}_2 \in \overline{\mathsf{HNN}}$. Indeed, f is computed by an mpPTF with $n+1$ terms:

$$\min\{L_1(\boldsymbol{x},\boldsymbol{y}), L_3(\boldsymbol{x},\boldsymbol{y}), \cdots\} \leq \min\{-1, L_2(\boldsymbol{x},\boldsymbol{y}), L_4(\boldsymbol{x},\boldsymbol{y}), \cdots\}$$

where $L_k(\boldsymbol{x},\boldsymbol{y}) = (k+1)\cdot(1-x_i-y_i)$. Note that if $x_i = y_i = 1$, then $L_i(\boldsymbol{x},\boldsymbol{y}) = -(i+1)$ and otherwise $L_i(\boldsymbol{x},\boldsymbol{y}) \geq 0$. Hence, the minimum is obtained at the maximum index j where $x_j = y_j = 1$. The claim follows from Theorem 14.

Second, it is known that $\mathsf{OMB} \circ \mathsf{AND}_2 \notin \mathsf{MAJ} \circ \mathsf{MAJ}$ by [4, 16]. Thus, if HNN was in $\mathsf{MAJ} \circ \mathsf{MAJ}$, then we could use the $\overline{\mathsf{HNN}}$ representation described above to get a $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuit computing $\mathsf{OMB} \circ \mathsf{AND}_2$, which is a contradiction. ◀

Finally, we observe a connection between mpPTFs and linear decision lists. This provides additional proof techniques for $\overline{\mathsf{HNN}}$ and helps to relate a question of separation of $\overline{\mathsf{HNN}}$ and $\overline{\mathsf{NN}}$ to the similar question for linear decision lists. The following lemma is proved in Appendix A.4.

▶ **Lemma 18.**

$$\mathsf{mpPTF}(\mathrm{poly}(n)) \subseteq \widehat{\mathsf{LDL}}.$$

More precisely, any mpPTF *with* m *terms and maximum weight* W *is equivalent to a linear decision list with length and maximum weight* $O(m^2 W)$.

▶ Remark 19. This lemma enables another technique to prove lower bounds for $\overline{\mathsf{HNN}}$ apart from sign-rank. More specifically, it is known that any function without large monochromatic rectangles must have a large linear decision list by [5].

▶ **Lemma 20.** $\mathsf{LDL} \subseteq \mathsf{mpPTF}(\infty)$.

Proof. It was shown in [18, Lemma 22] that $\mathsf{OMB} \circ \mathsf{THR} \subseteq \mathsf{mpPTF}(\infty)$. Our lemma follows since OMB is complete for the class of decision lists – See [18, Lemma 22]. ◀

It is open whether $\widehat{\mathsf{LDL}}$ and LDL are equal by [5]. Lemmas 18 and 20 immediately allow us to relate this problem to the problem of separating $\overline{\mathsf{HNN}}$ and $\overline{\mathsf{NN}}$.

▶ **Corollary 21.** *If* $\widehat{\mathsf{LDL}} \neq \mathsf{LDL}$, *then* $\overline{\mathsf{HNN}} \neq \overline{\mathsf{NN}}$.

Proof. From Theorem 14 and Lemmas 18, 20, we have the following sequence of inclusions.

$$\overline{\mathsf{HNN}} = \mathsf{mpPTF}(\mathrm{poly}) \subseteq \widehat{\mathsf{LDL}} \subseteq \mathsf{LDL} \subseteq \mathsf{mpPTF}(\infty) = \overline{\mathsf{NN}},$$

If $\overline{\mathsf{HNN}} = \overline{\mathsf{NN}}$, then the whole sequence of inclusions collapses and, in particular, $\widehat{\mathsf{LDL}} = \mathsf{LDL}$. ◀

4 kNN vs. Circuits

In this section, we give a circuit-style characterization of kNN and provide connections to known circuit classes. From these results, we obtain a separation between kNN and NN. Additionally, our results imply complexity theoretic barriers for proving superpolynomial lower bounds for kNN representations of explicit functions.

4.1 Characterization for small k

Here, we use the connection to mpPTF representations to get our first results on k-nearest neighbors complexity. In particular, we relate k-nearest neighbors representations for constant k to $\overline{\text{NN}}$ and prove a lower bound on k-nearest neighbors complexity for sublinear k.

▶ **Theorem 22.** *Any Boolean function with an m-anchor* kNN *representation is computed by an* mpPTF *with $\binom{m}{k}$ terms.*

Proof. We prove only the first statement as both arguments are identical. As noted in the proof of Theorem 14, the distances from anchors to a query point $\boldsymbol{x}$ are linear forms $L_1(\boldsymbol{x}), \cdots, L_m(\boldsymbol{x})$. Assign each linear form a label $\ell_1, \cdots, \ell_m \in \{1, -1\}$ where a positive label indicates placement on the left-hand side of the mpPTF and vice versa.

Then, consider the collection $A^+(\boldsymbol{x}) = \{L_{i_1}(\boldsymbol{x}) + \cdots + L_{i_k}(\boldsymbol{x}) \mid \ell_{i_1} + \cdots + \ell_{i_k} \geq 0\}$ and the compliment $A^-(\boldsymbol{x}) = \{L_{i_1}(\boldsymbol{x}) + \cdots + L_{i_k}(\boldsymbol{x}) \mid \ell_{i_1} + \cdots + \ell_{i_k} < 0\}$. The resulting mpPTF with $\binom{m}{k}$ terms, $\mathbb{1}[\min A^+(\boldsymbol{x}) \leq \min A^-(\boldsymbol{x})]$, realizes the original kNN representation: The minimum is attained by groups of k-nearest neighbors and if any such group has a positive majority then the inequality holds. ◀

It follows that Boolean functions with m-anchor kNN representations can be represented in $\overline{\text{NN}}$ with $\binom{m}{k}$ anchors. These results generalize to both weighted kNN and to non-Boolean inputs. See Appendix A.5 for a discussion.

As a consequence of Theorem 22, sign-rank lower bounds (e.g., Corollary 16) also apply to kNN. In particular, we get an exponential lower bound for kNN with $k = O(n^{1-\epsilon})$ for constant $\epsilon > 0$. This addresses an open question posed in [17] regarding k-nearest neighbors complexity.

▶ **Corollary 23.** *Any* $\overline{\text{kNN}}$ *representation of* IP *or $f_n(\boldsymbol{x}, \boldsymbol{y}) := \bigwedge_{i=1}^{n} \bigvee_{j=1}^{n^2} (x_{i,j} \wedge y_{i,j})$ requires $2^{\Omega(n/k)}$ anchors.*

Proof. Assume that IP (or f_n) has a $\overline{\text{kNN}}$ representation with m anchors. By Theorems 14 and 22 , IP has an $\overline{\text{NN}}$ representation with $\binom{m}{k} \leq m^k$ anchors. By Corollary 16, we have $m^k \geq 2^{\Omega(n)}$ and thus $m \geq 2^{\Omega(n/k)}$. ◀

4.2 Characterization for arbitrary k

In this section, we generalize the ideas of Theorem 14 to the closure of kNN, yielding further connections between nearest neighbors and circuit complexity.

▶ **Definition 24.** *Define by* kSTAT *the class of functions $f\colon \{0,1\}^n \to \{0,1\}$ representable by an inequality between k-statistics of two sets consisting of a polynomial number of linear forms: Given $\{L_1, \cdots, L_{\ell_1}\} \cup \{R_1, \cdots, R_{\ell_2}\}$ and integers k_l and k_r,*

$$f(\boldsymbol{x}) = 1 \iff (L_1(\boldsymbol{x}), \cdots, L_{\ell_1}(\boldsymbol{x}))_{(k_l)} < (R_1(\boldsymbol{x}), \cdots, R_{\ell_2}(\boldsymbol{x}))_{(k_r)} \tag{3}$$

and $\ell_1 + \ell_2$ is bounded by a polynomial in n.

As usual, we can assume that all coefficients in the linear forms are integers. Define the subclass $\widehat{\mathsf{kSTAT}}$ *where all coefficients are bounded by a polynomial in* n[7].

Note that we can reduce Definition 24 to the case of $k_l = k_r$ with only a linear increase in the size. This can be done by adding "dummy" linear forms that are always smaller than all others.

▶ **Theorem 25.**

$\overline{\mathsf{kNN}} = \mathsf{kSTAT}$, $\overline{\mathsf{kHNN}} = \widehat{\mathsf{kSTAT}}$.

See Appendix A.6 for the proof. Next, we provide another equivalent form of kSTAT that is sometimes more convenient.

▶ **Theorem 26.** *The class* kSTAT *consists exactly of functions* $f\colon \{0,1\}^n \to \{0,1\}$ *for which there exist linear forms* $\{L_1, \cdots, L_p\}$ *with* $p = \mathrm{poly}(n)$*, a positive integer* k*, and a labelling function* $\mathsf{label}\colon \{1, \cdots, p\} \to \{0,1\}$*, such that for all* $\boldsymbol{x}$*,*

$$f(\boldsymbol{x}) = 1 \iff (L_1(\boldsymbol{x}), \cdots, L_p(\boldsymbol{x}))_{(k)} = L_i(\boldsymbol{x}) \text{ for some } i \text{ with } \mathsf{label}(i) = 1. \tag{4}$$

The class $\widehat{\mathsf{kSTAT}}$ *consists exactly of functions with the same representation with polynomial-size coefficients in the linear forms.*

See Appendix A.7 for the proof. Now we show that some well-known circuit classes, for which we do not have any known lower bounds, are computable by $\overline{\mathsf{kHNN}}$.

▶ **Theorem 27.**

$\mathsf{SYM} \circ \mathsf{MAJ} \subseteq \widehat{\mathsf{kSTAT}}$.

Any symmetric function of s *threshold functions has a* $\widehat{\mathsf{kSTAT}}$ *representation with* $k = s + 1$.

See Appendix A.8 for the proof. Using the same strategy, we can embed a large complexity class into kNN directly:

▶ **Theorem 28.**

$\mathsf{SYM} \circ \mathsf{AND} \subseteq \mathsf{kNN}$.

Any symmetric function of s *conjunctions has a* kNN *representation with* $k = 2s + 1$.

See Appendix A.9 for the proof.

▶ Remark 29. Note that $\mathsf{SYM} \circ \mathsf{AND} \subseteq \mathsf{SYM} \circ \mathsf{MAJ}$ and $\mathsf{SYM} \circ \mathsf{AND}$ is known to simulate the whole class of ACC^0 within quasi-polynomial size [3]. Related classes are of interest in the context of obtaining lower bounds through circuit satisfiability algorithms [40, Conjecture 1].

As a result of Theorem 28, if we prove for some explicit function f that $f \notin \mathsf{kNN}$, it will follow that $f \notin \mathsf{SYM} \circ \mathsf{AND}$, and this would be a major breakthrough in circuit complexity. Also note that $\mathsf{IP} \in \mathsf{SYM} \circ \mathsf{AND}$ and thus, by Theorem 28, $\mathsf{IP} \in \mathsf{kNN}$. Together with Corollary 16, this gives a separation between NN and kNN. This also shows that in Corollary 23 we cannot get rid of k in the lower bound.

▶ **Theorem 30.** $\mathsf{ELDL} \subseteq \mathsf{kSTAT}$, $\widehat{\mathsf{ELDL}} \subseteq \widehat{\mathsf{kSTAT}}$.

See Appendix A.10 for the proof.

▶ Remark 31. The class ELDL is known to be contained in $\mathsf{THR} \circ \mathsf{THR}$ and proving super-polynomial lower bounds for ELDL is an open problem (See [9]).

[7] mpPTF can be viewed as a special case of kSTAT in which $k_l = k_r = 1$.

5 New bounds for the nearest neighbor complexity of Boolean functions

In this section, we derive several bounds on the nearest neighbor complexity of Boolean functions.

5.1 Nearest neighbor complexity of CNFs

We first show that any CNF admits an efficient NN representation.

▶ **Theorem 32.** *Any CNF or DNF with m clauses has an* NN *representation with $m+1$ anchors and constant bit-complexity.*

Proof. It suffices to prove the statement for DNFs as any CNF can be converted to a DNF by negation.

Let $N = \{\boldsymbol{q} := (\frac{1}{2}, \cdots, \frac{1}{2})\}$ and note that $d(\boldsymbol{x}, \boldsymbol{q}) = n/4$ for every input $\boldsymbol{x} \in \{0,1\}^n$ (where d is the squared Euclidean distance). For each clause, say $C(\boldsymbol{x}) = (x_1 \wedge \cdots \wedge x_k)$, introduce a positive anchor

$$\boldsymbol{p_C} = \left(\underbrace{1, \frac{3}{2}, \cdots, \frac{3}{2}}_{k}, \underbrace{\frac{1}{2}, \cdots, \frac{1}{2}}_{n-k}\right)$$

If any variable is negated, replace the corresponding $\frac{3}{2}$ (or 1) with $-\frac{1}{2}$ (or 0).

If $C(\boldsymbol{x}) = 1$, then $d(\boldsymbol{x}, \boldsymbol{p_C}) = (n-1)/4 < d(\boldsymbol{x}, \boldsymbol{q})$. Otherwise, some literal in C is equal to zero, hence $d(\boldsymbol{x}, \boldsymbol{p_C}) \geq 1 + (n-1)/4 > d(\boldsymbol{x}, \boldsymbol{q})$. Therefore, the entire DNF, say $C_1 \vee \cdots \vee C_m$, is satisfied if and only if some $\boldsymbol{p_{C_i}}$ is a nearest neighbor of $\boldsymbol{x}$. ◀

The polynomial-size representation above does not generalize to deeper AC^0 circuits of depth larger than 2. For instance, Corollary 16 exhibits a function computable by a depth-three De Morgan circuit of polynomial size which does not belong to $\overline{\mathsf{NN}}$. For the well studied disjointness function (that admits a compact CNF representation) we can get an efficient HNN representation:

▶ **Theorem 33.**

$$\mathsf{DISJ} \in \mathsf{HNN}$$

The disjointness function (in $2n$ dimensions) has an HNN *representation with $3n$ anchors.*

Proof. Consider anchors $P = \{(\boldsymbol{e_1}, \boldsymbol{e_1}), \cdots, (\boldsymbol{e_n}, \boldsymbol{e_n})\}$ and $N = \{\boldsymbol{e_1}, \cdots, \boldsymbol{e_{2n}}\}$ where $\boldsymbol{e_i}$ denotes the i'th standard basis vector and $(\boldsymbol{e_i}, \boldsymbol{e_i})$ their concatenation.

Let $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^n$ and suppose $x_i = y_i = 1$ for some i. Then, for all j it holds that $\Delta((\boldsymbol{x}, \boldsymbol{y}), (\boldsymbol{e_i}, \boldsymbol{e_i})) \leq \Delta((\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{e_j}) - 1$ with equality when $i = j$. Otherwise, $\Delta((\boldsymbol{x}, \boldsymbol{y}), (\boldsymbol{e_i}, \boldsymbol{e_i})) \geq \Delta((\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{e_j}) + 1$ for all i, j. ◀

▶ Remark 34. It can be shown that the number of anchors in Theorem 33 is nearly tight; based on the $\Omega(n)$ lower bound for DISJ of [33], a simple argument proves that NN representations of disjointness require $\Omega(n/\log n)$ anchors. We omit the details.

Proceeding, we show that some CNFs with polynomially many clauses have exponential *Boolean* nearest neighbor complexity.

▶ **Definition 35.** *The* Hamming cube graph *is an undirected graph with vertices* $V = \{0,1\}^n$ *and edges* $E = \{(\boldsymbol{u}, \boldsymbol{v}) \in V : \Delta(\boldsymbol{u}, \boldsymbol{v}) = 1\}$. *The* components *of a Boolean function* f *are the connected components of the subgraph of the Hamming cube graph induced by the vertex set* $f^{-1}(1)$.

▶ **Lemma 36.** *If a Boolean function* f *has* m *components then any* HNN *representation of* f *has at least* m *anchors.*

Proof. Consider some component C of f and let $\delta(C)$ denote the vertex boundary of C: Vertices in $\{0,1\}^n \setminus C$ with a neighbor in C. Note that $\delta(C) \subseteq f^{-1}(0)$.

Suppose f has HNN representation $P \cup N$ and let $\boldsymbol{p} \in P$ be the nearest anchor to some $\boldsymbol{x} \in C$. Assume for contradiction that $\boldsymbol{p} \notin C$. Note that $\Delta(\boldsymbol{x}, \boldsymbol{p})$ is equal to the length of the shortest path from $\boldsymbol{x}$ to $\boldsymbol{p}$ in the Hamming cube graph, which by assumption must contain some $\boldsymbol{y} \in \delta(C)$. (In particular, $\Delta(\boldsymbol{x}, \boldsymbol{p}) = \Delta(\boldsymbol{x}, \boldsymbol{y}) + \Delta(\boldsymbol{y}, \boldsymbol{p})$.) Thus, there must exist some negative anchor $\boldsymbol{q} \in N$ with $\Delta(\boldsymbol{y}, \boldsymbol{q}) < \Delta(\boldsymbol{y}, \boldsymbol{p})$. By the triangle inequality,

$$\Delta(\boldsymbol{x}, \boldsymbol{q}) \leq \Delta(\boldsymbol{x}, \boldsymbol{y}) + \Delta(\boldsymbol{y}, \boldsymbol{q}) < \Delta(\boldsymbol{x}, \boldsymbol{y}) + \Delta(\boldsymbol{y}, \boldsymbol{p}) = \Delta(\boldsymbol{x}, \boldsymbol{p})$$

which contradicts the minimality of $\boldsymbol{p}$. Thus, each component contains an anchor. ◀

Using the previous results, another separation between HNN and NN follows from the existence of a CNF (over n-variables) with $\text{poly}(n)$ clauses and exponentially (in n) many components. (See Appendix A.)

▶ **Theorem 37.** *For any* $k > 0$*, there exists a* k*-CNF over* n *variables with* $\text{poly}(n)$ *clauses for which any* HNN *representation has* $2^{\Omega(n)}$ *anchors.*

5.2 A new lower bound for majority

We now discuss the disparity between the HNN complexity of the majority function in [17, Theorem 4]: In particular, when n is even, the best upper bound is $\frac{n}{2} + 2$ anchors, whereas 2 anchors suffices when n is odd. Note that if ties were allowed (won by positive anchors) in Definition 3, then $P = \{1^n\}$ and $N = \{0^n\}$ would suffice as an HNN representation for MAJ for all n.

▶ **Theorem 38.** *For even* n*, any* HNN *representation of* MAJ *requires* $\frac{n}{2} + 2$ *anchors.*

Proof. Suppose $P \cup N$ is an HNN representation of MAJ for even n. We claim that for each $\boldsymbol{x} \in \{0,1\}^n$ satisfying $\Delta(\boldsymbol{x}) = n/2$, there is a positive anchor $\boldsymbol{p} \neq \mathbf{1}$ with $\boldsymbol{x} \leq \boldsymbol{p}$ in coordinate-wise order:

It follows from [17] that the nearest anchor $\boldsymbol{p}$ to $\boldsymbol{x}$ satisfies $\boldsymbol{x} \leq \boldsymbol{p}$. Indeed, for some i it holds that $x_i = 1$, so suppose for contradiction that $p_i = 0$. Then, construct $\boldsymbol{y} = \boldsymbol{x} - \boldsymbol{e_i}$ and let $\boldsymbol{q} \in N$ be the nearest anchor to $\boldsymbol{y}$. This yields $\Delta(\boldsymbol{x}, \boldsymbol{p}) = \Delta(\boldsymbol{y}, \boldsymbol{p}) + 1 > \Delta(\boldsymbol{y}, \boldsymbol{q}) + 1$, contradicting the fact that

$$\Delta(\boldsymbol{x}, \boldsymbol{p}) < \Delta(\boldsymbol{x}, \boldsymbol{q}) \leq \Delta(\boldsymbol{y}, \boldsymbol{q}) + 1. \tag{5}$$

A similar argument shows that $\boldsymbol{q} \leq \boldsymbol{y}$. Hence, $\Delta(\boldsymbol{y}, \boldsymbol{q}) \leq \frac{n}{2} - 1$, and (5) becomes $\Delta(\boldsymbol{x}, \boldsymbol{p}) < \frac{n}{2}$ which implies that $\Delta(\boldsymbol{p}) \leq n - 1$, proving the claim.

For contradiction, assume that $|P \cup N| \leq \frac{n}{2} + 1$. Since there must be at least one negative anchor, we have $|P| \leq \frac{n}{2}$. Then, we can construct $\boldsymbol{x} \in \{0,1\}^n$ with $\Delta(\boldsymbol{x}) = \frac{n}{2}$ for which there is no positive anchor $\boldsymbol{p} \neq \mathbf{1}$ with $\boldsymbol{x} \leq \boldsymbol{p}$, leading to a contradiction: For each $\boldsymbol{p} \in P \setminus \mathbf{1}$, arbitrarily select some i where $p_i = 0$ and set $x_i = 1$, ensuring $\boldsymbol{x} \not\leq \boldsymbol{p}$. After this process, $\Delta(\boldsymbol{x}) \leq |P| \leq \frac{n}{2}$. Arbitrarily fixing more coordinates of $\boldsymbol{x}$ to 1 so that $\Delta(\boldsymbol{x}) = \frac{n}{2}$ completes the construction. ◀

6 Conclusion

We have studied nearest neighbor representations of Boolean functions, proving new lower and upper bounds and devising connections to circuit complexity. There are many future questions and research directions:

- Studying representations of Boolean functions using ideas from *approximate* nearest neighbor search [20, 28] could be of interest. Such a study could potentially lead to new insights and more compact representations avoiding the curse of dimensionality.
- Studying nearest neighbor complexity with respect to additional discrete domains such as grids as well as more than two labels is an interesting future direction.
- Circuit complexity has been used to derive new algorithms for nearest neighbor problems [1]. Can ideas about nearest neighbor complexity such as connections to mpPTFs be used to obtain new algorithms for nearest neighbor classification and search?
- Finally, it remains open whether $\overline{\mathsf{NN}} = \overline{\mathsf{HNN}}$.

References

1 Josh Alman and Ryan Williams. Probabilistic polynomials and hamming nearest neighbors. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 136–150. IEEE, 2015. doi:10.1109/FOCS.2015.18.

2 Alexandr Andoni. *Nearest neighbor search: the old, the new, and the impossible*. PhD thesis, Massachusetts Institute of Technology, 2009.

3 Richard Beigel and Jun Tarui. On ACC. *Comput. Complex.*, 4:350–366, 1994. doi:10.1007/BF01263423.

4 Harry Buhrman, Nikolay Vereshchagin, and Ronald de Wolf. On computation and communication with small bias. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*, pages 24–32. IEEE, 2007. doi:10.1109/CCC.2007.18.

5 Arkadev Chattopadhyay, Meena Mahajan, Nikhil S. Mande, and Nitin Saurabh. Lower bounds for linear decision lists. *Chic. J. Theor. Comput. Sci.*, 2020, 2020. URL: http://cjtcs.cs.uchicago.edu/articles/2020/1/contents.html.

6 Yan Qiu Chen, Mark S Nixon, and Robert I Damper. Implementing the k-nearest neighbour rule via a neural network. In *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 1, pages 136–140. IEEE, 1995. doi:10.1109/ICNN.1995.488081.

7 Kenneth L Clarkson. Nearest neighbor queries in metric spaces. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 609–617, 1997. doi:10.1145/258533.258655.

8 Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967. doi:10.1109/TIT.1967.1053964.

9 Yogesh Dahiya, K. Vignesh, Meena Mahajan, and Karteek Sreenivasaiah. Linear threshold functions in decision lists, decision trees, and depth-2 circuits. *Inf. Process. Lett.*, 183:106418, 2024. doi:10.1016/J.IPL.2023.106418.

10 Luc Devroye. On the asymptotic probability of error in nonparametric discrimination. *The Annals of Statistics*, 9(6):1320–1327, 1981.

11 Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition*, volume 31. Springer Science & Business Media, 2013.

12 Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In *Conference on learning theory*, pages 907–940. PMLR, 2016. URL: http://proceedings.mlr.press/v49/eldan16.html.

13 Jürgen Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *Journal of Computer and System Sciences*, 65(4):612–625, 2002. doi:10.1016/S0022-0000(02)00019-3.

14 Mikael Goldmann, Johan Håstad, and Alexander Razborov. Majority gates vs. general weighted threshold gates. *Computational Complexity*, 2:277–300, 1992. doi:10.1007/BF01200426.

15 Mikael Goldmann and Marek Karpinski. Simulating threshold circuits by majority circuits. *SIAM Journal on Computing*, 27(1):230–246, 1998. doi:10.1137/S0097539794274519.

16 András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 46(2):129–154, 1993. doi:10.1016/0022-0000(93)90001-D.

17 Péter Hajnal, Zhihao Liu, and György Turán. Nearest neighbor representations of boolean functions. *Information and Computation*, 285:104879, 2022. doi:10.1016/J.IC.2022.104879.

18 Kristoffer Arnsfelt Hansen and Vladimir V Podolskii. Polynomial threshold functions and boolean threshold circuits. *Information and Computation*, 240:56–73, 2015. doi:10.1016/J.IC.2014.09.008.

19 Lisa Hellerstein and Rocco A Servedio. On PAC learning algorithms for rich boolean function classes. *Theoretical Computer Science*, 384(1):66–76, 2007. doi:10.1016/J.TCS.2007.05.018.

20 Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998. doi:10.1145/276698.276876.

21 Piotr Indyk and Tal Wagner. Approximate nearest neighbors in limited space. In *Conference On Learning Theory*, pages 2012–2036. PMLR, 2018. URL: http://proceedings.mlr.press/v75/indyk18a.html.

22 Jeffrey C Jackson, Adam R Klivans, and Rocco A Servedio. Learnability beyond AC^0. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 776–784, 2002.

23 Stasys Jukna. *Boolean function complexity: advances and frontiers*, volume 5. Springer, 2012. doi:10.1007/978-3-642-24508-4.

24 Kordag Mehmet Kilic, Jin Sima, and Jehoshua Bruck. On the information capacity of nearest neighbor representations. In *2023 IEEE International Symposium on Information Theory (ISIT)*, pages 1663–1668, 2023. doi:10.1109/ISIT54713.2023.10206832.

25 Kordag Mehmet Kilic, Jin Sima, and Jehoshua Bruck. Nearest neighbor representations of neurons. In *2024 IEEE International Symposium on Information Theory (ISIT)*, 2024.

26 Adam R Klivans and Rocco A Servedio. Learning DNF in time $2^{O(n^{1/3})}$. *Journal of Computer and System Sciences*, 2(68):303–318, 2004.

27 Adam R Klivans and Rocco A Servedio. Toward attribute efficient learning of decision lists and parities. *Journal of Machine Learning Research*, 7(4), 2006. URL: https://jmlr.org/papers/v7/klivans06a.html.

28 Eyal Kushilevitz, Rafail Ostrovsky, and Yuval Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 614–623, 1998. doi:10.1145/276698.276877.

29 Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. *Journal of the ACM (JACM)*, 40(3):607–620, 1993. doi:10.1145/174130.174138.

30 James Martens, Arkadev Chattopadhya, Toni Pitassi, and Richard Zemel. On the representational efficiency of restricted boltzmann machines. *Advances in Neural Information Processing Systems*, 26, 2013.

31 O Murphy. Nearest neighbor pattern classification perceptrons. *Neural Networks: Theoretical Foundations and Analysis*, pages 263–266, 1992.

32 Edward A Patrick and Frederic P Fischer III. A generalized k-nearest neighbor rule. *Information and control*, 16(2):128–152, 1970. doi:10.1016/S0019-9958(70)90081-1.

33 Alexander A Razborov. On the distributional complexity of disjointness. In *International Colloquium on Automata, Languages, and Programming*, pages 249–253. Springer, 1990. doi:10.1007/BFB0032036.

34 Alexander A Razborov. On small depth threshold circuits. In *Scandinavian Workshop on Algorithm Theory*, pages 42–52. Springer, 1992. doi:10.1007/3-540-55706-7_4.

35 Alexander A Razborov and Alexander A Sherstov. The sign-rank of AC^0. *SIAM Journal on Computing*, 39(5):1833–1855, 2010.

36 Ronald L Rivest. Learning decision lists. *Machine learning*, 2:229–246, 1987. doi:10.1007/BF00058680.

37 Michael E. Saks. *Slicing the hypercube*, pages 211–256. London Mathematical Society Lecture Note Series. Cambridge University Press, 1993.

38 Matus Telgarsky. Benefits of depth in neural networks. In *Conference on learning theory*, pages 1517–1539. PMLR, 2016. URL: http://proceedings.mlr.press/v49/telgarsky16.html.

39 Gal Vardi, Daniel Reichman, Toniann Pitassi, and Ohad Shamir. Size and depth separation in approximating benign functions with neural networks. In *Conference on Learning Theory*, pages 4195–4223. PMLR, 2021. URL: http://proceedings.mlr.press/v134/vardi21a.html.

40 Nikhil Vyas and R. Ryan Williams. Lower bounds against sparse symmetric functions of ACC circuits: Expanding the reach of #SAT algorithms. *Theory Comput. Syst.*, 67(1):149–177, 2023. doi:10.1007/S00224-022-10106-8.

41 R. Ryan Williams. Limits on representing boolean functions by linear combinations of simple functions: Thresholds, relus, and low-degree polynomials. In *33rd Computational Complexity Conference (CCC 2018)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018.

A Omitted proofs

A.1 Proof of Theorem 14

We break the proof of this theorem into two separate lemmas.

▶ **Lemma 39.**

$$\overline{\mathsf{NN}} \subseteq \mathsf{mpPTF}(\infty), \quad \overline{\mathsf{HNN}} \subseteq \mathsf{mpPTF}(\mathrm{poly}(n))$$

More precisely, any $\overline{\mathsf{NN}}$ *representation with* m *anchors is equivalent to an* mpPTF *with* m *terms, and any* $\overline{\mathsf{HNN}}$ *representation with* m *anchors in* $\widetilde{n}$ *dimensions is equivalent to an* mpPTF *with* m *terms and maximum weight* $\widetilde{n}$.

Proof. The distance from $\boldsymbol{x} \in \{0,1\}^n$ to an anchor $\boldsymbol{p} \in \mathbb{R}^n$ is a linear form in variables $\boldsymbol{x}$:

$$\begin{aligned}\sum_i (x_i - p_i)^2 &= \sum_i \left[x_i^2 - 2p_i x_i + p_i^2\right] \\ &= \sum_i \left[(1-2p_i)x_i + p_i^2\right] \\ &= \langle \mathbf{1} - 2\boldsymbol{p}, \boldsymbol{x}\rangle + ||\boldsymbol{p}||_2^2.\end{aligned}$$

We can observe that NN representations essentially compute $\mathbb{1}[\min_{\boldsymbol{p}\in P} \Delta(\boldsymbol{x},\boldsymbol{p}) \leq \min_{\boldsymbol{q}\in N} \Delta(\boldsymbol{x},\boldsymbol{q})]$, which is an mpPTF. Subfunctions merely multiply coefficients and add constants to each linear form – For example, $d(x_1x_10, p_1p_2p_3) = 2\cdot(1-2p_1)x_1 + (p_3^2 + 2p_1^2)$.

In the case of HNN, we have for all anchors that $\boldsymbol{p} \in \{0,1\}^n$ and $\Delta(\boldsymbol{x},\boldsymbol{p})$ is a linear form with ± 1 coefficients and positive constants bounded (in absolute value) by n. As a result, the weights in mpPTF are bounded by n as well. ◀

▶ **Lemma 40.**

$$\mathsf{mpPTF}(\infty) \subseteq \overline{\mathsf{NN}}, \quad \mathsf{mpPTF}(\mathrm{poly}(n)) \subseteq \overline{\mathsf{HNN}}$$

More precisely, any mpPTF *with* m *terms has an* $\overline{\mathsf{NN}}$ *representation with* m *anchors in* $n+1$ *dimensions. Any* mpPTF *with* m *terms and maximum weight* W *has an* $\overline{\mathsf{HNN}}$ *representation with* m *anchors in* $\widetilde{n} = O(nW)$ *dimensions.*

Proof. We start with the $\mathsf{mpPTF}(\mathrm{poly}(n))$ case. Let $\mathbb{1}[\min_{i\le\ell_1} L_{1i}(\boldsymbol{x}) \le \min_{j\le\ell_2} L_{2j}(\boldsymbol{x})]$ be an arbitrary $\mathsf{mpPTF}(\mathrm{poly}(n))$. First make some pre-processing steps. First, multiply each linear form by 2 and add one to the right-hand side, so that ties are won by the left-hand side. Second, we would like to make all coefficients positive. For this, while there exists a negative term $-a_{ijk}x_k$ (or constant $-\theta_{ij}$), just add x_k (or 1) to every linear form until all negative terms are eliminated. No coefficient (or constant) will increase by more than W. Third, we make all coefficients even by multiplying all linear forms by two. Finally, we add the same constant Θ (to be decided later) to all linear forms. Then, every linear form is equal to $L_{ij}(\boldsymbol{x}) = a_{ij1}x_1 + \cdots + a_{ijn}x_n + \theta_{ij} + \Theta$, for positive, even constants $a_{ijk}, \theta_{ij} \le 8W$.

Define n *block sizes* $t_1, \cdots, t_n$ by $t_k := \max_{i,j} a_{ijk}$ (i.e., the maximum coefficient of x_k in any linear form). Also define $C = \Theta + \max_{i,j}\theta_{ij}$ and let $\widetilde{n} := t_1 + \cdots + t_n + C$. Inputs $\boldsymbol{x} \in \{0,1\}^n$ are mapped ($\rightsquigarrow$) to query points $\widetilde{\boldsymbol{x}} \in \{0,1\}^{\widetilde{n}}$ and linear forms L_{ij} are mapped to anchors $\widetilde{\boldsymbol{p}}_{ij} \in \{0,1\}^{\widetilde{n}}$ such that $\Delta(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{p}}_{ij}) = L_{ij}(\boldsymbol{x})$. In particular,

$$\widetilde{\boldsymbol{x}} := \underbrace{x_1 \cdots\cdots x_1}_{t_1 \text{ many}} \cdots \underbrace{x_n \cdots\cdots x_n}_{t_n \text{ many}} \cdot \underbrace{1 \cdots 1}_{C \text{ many}}$$

and

$$\widetilde{\boldsymbol{p}}_{ij} = \underbrace{0\cdots\cdots 0}_{(t_1+a_{ij1})/2}\,\underbrace{1\cdots\cdots 1}_{(t_1-a_{ij1})/2} \cdots \underbrace{0\cdots\cdots 0}_{(t_n+a_{ijn})/2}\,\underbrace{1\cdots\cdots 1}_{(t_n-a_{ijn})/2} \cdot \underbrace{0\cdots\cdots 0}_{z_{ij}}\underbrace{1\cdots\cdots 1}_{C-z_{ij}}$$

where z_i will be chosen momentarily. (Let $P = \{\widetilde{\boldsymbol{p}}_{1j}\}_{j\le\ell_1}$ and $N = \{\widetilde{\boldsymbol{p}}_{2j}\}_{j\le\ell_2}$.) The distance between $\widetilde{\boldsymbol{x}}$ and $\widetilde{\boldsymbol{p}}_{ij}$ is equal to

$$\begin{aligned}\Delta(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{p}}_{ij}) &= z_{ij} + \sum_k \left(\frac{t_k + a_{ijk}}{2}\right) x_k + \left(\frac{t_k - a_{ijk}}{2}\right)(1 - x_k) \\ &= z_{ij} + \langle \boldsymbol{a_{ij}}, \boldsymbol{x}\rangle + \sum_k \left(\frac{t_k - a_{ijk}}{2}\right)\end{aligned}$$

Now let $z_{ij} = \Theta + \theta_{ij} - \sum_k \left(\frac{t_k - a_{ijk}}{2}\right)$ so that $\Delta(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{p}}_{ij}) = \langle \boldsymbol{a_{ij}}, \boldsymbol{x}\rangle + \Theta + \theta_{ij}$. This is valid (i.e., z_{ij} is a non-negative integer) if we choose a large enough value for Θ: The minimal value of Θ such that $z_{ij} \ge 0$ for all i, j is

$$\Theta = \max_{i,j}\left(\sum_k \left(\frac{t_k - a_{ijk}}{2}\right) - \theta_{ij}\right) \le \sum_k \frac{t_k}{2} \le 4nW.$$

Thus, for $\Theta = 4nW$, we may always choose $0 \le z_{ij} \le \Theta + \theta_{ij} \le C$. Observe that $\boldsymbol{x} \rightsquigarrow \widetilde{\boldsymbol{x}}$ by duplicating each x_i at most $8W$ times and introducing at most $4nW + 8W$ constant variables. Thus, the original mpPTF is equivalent to a subfunction of an HNN representation with m anchors at most $4nW + 8W$ dimensions.

For the $\mathsf{mpPTF}(\infty)$ case, the same method applies, only now we do not need to increase the dimension that much. All coefficients can be realized by choosing anchors $\boldsymbol{p}_{ij} = (1 - \boldsymbol{a}_{ij})/2$ and all constants θ_{ij} can be corrected using one additional dimension. ◀

From this we can also deduce the following:

▶ **Theorem 41.** *Any function with an m-anchor NN representation with bit-complexity $O(\log n)$ is equivalent to an $\mathsf{mpPTF}(\mathrm{poly}(n))$ with m terms. Any function of n inputs with an $\mathsf{mpPTF}(\mathrm{poly}(n))$ representation with m terms is equivalent to a subfunction of a function of $n+1$ inputs with an m-anchor NN representation with bit-complexity $O(\log n)$.*

Proof. Observe that in Lemmas 39 and 40 – for NN and $\mathsf{mpPTF}(\infty)$ – the bit-complexity of NN and the logarithms of weights of mpPTF are linearly related. ◀

A.2 Proof of Corollary 15

Proof. It is shown in [17] that XOR has a unique HNN representation with 2^n anchors. Furthermore, it is established in [18] that $\mathsf{XOR} \in \mathsf{mpPTF}(\mathrm{poly}(n))$: In particular, $\mathsf{XOR}(\boldsymbol{x}) = 1$ if and only if $\min\{L_0(\boldsymbol{x}), L_2(\boldsymbol{x}), \cdots\} \leq \min\{L_1(\boldsymbol{x}), L_3(\boldsymbol{x}), \cdots\}$ where $L_i(\boldsymbol{x}) = i^2 - 2i \cdot (x_1 + \cdots + x_n)$. ◀

A.3 Proof of Corollary 16

Proof. It was shown by [17] that the NN complexity of a Boolean function f is bounded below by the *sign-rank* of f, and this can be easily extended to $\overline{\mathsf{NN}}$ through Theorem 14: The number of terms in an mpPTF computing f is also bounded below by the sign-rank of f, by [18].

[13] and [35] respectively establish that the sign rank of IP is equal to $2^{n/2}$ and the sign rank of f_n is $2^{\Omega(n)}$. ◀

A.4 Proof of Lemma 18

Proof. Consider a function $f \in \mathsf{mpPTF}(\mathrm{poly}(n))$ and let $\mathbb{1}[\min_{i \leq \ell_1} L_{1i}(\boldsymbol{x}) \leq \min_{j \leq \ell_2} L_{2j}(\boldsymbol{x})]$ be its representation. We can assume that all possible values of all linear forms are distinct. For this it is enough to multiply all forms by $\ell_1 + \ell_2$ and to add to each form it's own unique remainder modulo $\ell_1 + \ell_2$.

Observe that all linear forms obtain only polynomially many variables (since there output is polynomially bounded in absolute value). Denote possible values of the form L_{ij} by $a_{ij1}, \cdots, a_{ijt}$ for some t polynomially bounded in n. Note that, for different linear forms, the number of the values obtained might be not the same. To simplify the notation we assume that we add several equal values to the list to make them all of equal size t.

Now we are ready to produce the decision list. Let $c_1 = 1$ and $c_2 = 0$. We consider each a_{ijk} in increasing order and query if $L_{ij}(\boldsymbol{x}) \leq a_{ijk}$. If so, we output c_i. If not, we proceed to the next a_{ijk}.

This decision list computes f since we are just looking for the minimal value of a linear form among all possible values of the forms. ◀

A.5 Consequences of Theorem 22

▶ **Corollary 42.** *Any Boolean function with a* $\overline{\mathsf{kNN}}$ *representation with* m *anchors has an* $\overline{\mathsf{NN}}$ *representation with* $\binom{m}{k}$ *anchors. (Similarly, Boolean function with a* $\overline{\mathsf{kHNN}}$ *representation with* m *anchors has an* $\overline{\mathsf{HNN}}$ *representation with* $\binom{m}{k}$ *anchors.)*

▶ Remark 43. Theorem 22 and Corollary 42 can be extended to non-Boolean inputs. More precisely, the same statements are true over any finite domain $D \subseteq \mathbb{R}^n$. For this we can express (squared) distances to anchors as quadratic forms, for each subset of distances of size k consider the average of these distances and represent them as a distance to a new anchor. We still need to add an extra dimension to absorb constant terms.

▶ Remark 44. Theorem 22 and Corollaries 42 and 23 can be extended to the case of weighted kNN. Indeed, in Theorem 22, instead of sums of linear forms we will have weighted sums. This will require $\binom{m}{k} \cdot k! = \frac{m!}{(m-k)!}$ terms in the mpPTF representation. If the weights in the weighted kNN representation are small and the bit-complexity of anchors is small, this results in a $\overline{\mathsf{HNN}}$ representation and if there are no restrictions of weights and bit-complexity, we get $\overline{\mathsf{NN}}$ representation. The proof of Corollary 23 still works despite the increase of the number of anchors to $\frac{m!}{(m-k)!}$.

A.6 Proof of Theorem 25

We first make the following general observation: [32] show that finding the k'th nearest positive anchor and k'th nearest negative anchor and classifying based on which is closest is equivalent to computing a $(2k-1)$-nearest neighbors representation. This fact can be generalized, considering the closure of kNN.

▶ **Lemma 45.** *Let A and B be two sets of numbers and let S be the k smallest elements of $A \cup B$. Then,*

$$|A \cap S| \geq |B \cap S| \iff A_{(t)} < B_{(t)}$$

where $t = \left\lfloor \frac{k+1}{2} \right\rfloor$. (As in kNN, we assume S exists and is unique).

Proof. A contains a majority of the elements in S if and only if $|A \cap S| \geq t$. This happens if and only if the t'th smallest element in A is smaller than the t'th smallest element in B. ◀

We now proceed with the proof of Theorem 25.

Proof. For the inclusion $\overline{\mathsf{kNN}} \subseteq \mathsf{kSTAT}$, consider any function f in $\overline{\mathsf{kNN}}$. It is a subfunction of some function g with a kNN representation $P \cup N$. As in Lemma 39, the distances between $\boldsymbol{x}$ and each anchor are linear forms $A = \{L_1(\boldsymbol{x}), \cdots, L_{|P|}(\boldsymbol{x})\}$ and $B = \{R_1(\boldsymbol{x}), \cdots, R_{|N|}(\boldsymbol{x})\}$ which we assume have integer coefficients by the usual finite precision argument. By definition $g(\boldsymbol{x}) = 1$ if and only if the set S of k-nearest neighbors satisfies $|P \cap S| \geq |N \cap S|$. By Lemma 45, this happens if and only if $A_{(t)} < B_{(t)}$, taking $t = \left\lfloor \frac{k+1}{2} \right\rfloor$. Hence, $g \in \mathsf{kSTAT}$. As kSTAT is closed under taking subfunctions, $f \in \mathsf{kSTAT}$ as well.

For the inclusion $\mathsf{kSTAT} \subseteq \overline{\mathsf{kNN}}$, assume that f has a kSTAT representation. By adding dummy linear forms we can have $k_l = k_r$. By Lemma 45, the inequality (3) holds if and only if the $2k_l - 1$ smallest linear forms consist of more linear forms from the left-hand side than the right. Representing each inequality by an anchor, we obtain a representation of the same function in $\overline{\mathsf{kNN}}$.

The case of $\overline{\mathsf{kHNN}}$ and $\widehat{\overline{\mathsf{kSTAT}}}$ is analogous. ◀

A.7 Proof of Theorem 26

Proof. Suppose a Boolean function f has a representation $\{L_1, \cdots, L_p\}$ satisfying (4) for some function label and integer k. We will show that $f \in \mathsf{kSTAT}$. First, we assume that all coefficients in all linear form are integers and ensure that all values of all linear forms are distinct and even. For this, multiply all forms by $2p$ and shift each form by its own even remainder modulo $2p$.

For each $i \leq p$, we add one linear form to each side of (3). If $\mathsf{label}(i) = 1$, then place the form $L_i(\boldsymbol{x})$ on the left-hand side and $L_i(\boldsymbol{x}) + 1$ on the right. If $\mathsf{label}(i) = 0$, put the $L_i(\boldsymbol{x})$ on right-hand side and $L_i(\boldsymbol{x}) + 1$ on the left. It is easy to see that the k'th statistics in the left and right-hand sides of the resulting kSTAT representation are $L_i(\boldsymbol{x})$ and $L_i(\boldsymbol{x}) + 1$ (not necessarily in that order), where $L_i(\boldsymbol{x})$ is the k'th statistic of the original representation. Hence, the inequality in (3) holds if and only if $\mathsf{label}(i) = 1$.

For the other direction, assume we have a function $f \in \mathsf{kSTAT}$ given by (3). We again assume that all coefficients are integers and all values of all linear forms are distinct. Now we construct the required representation of f. For each form L_i we add to the representation the forms $L_{ij}(\boldsymbol{x}) := L_i(\boldsymbol{x}) + \frac{j}{k_l + k_r}$ for all $j \in \{0, 1, \cdots, k_l + k_r - 1\}$, and for each form R_i we add to the representation the forms $R_{ij}(\boldsymbol{x}) := R_i(\boldsymbol{x}) + \frac{j}{k_l + k_r + 1}$ for all $j = \{0, 1, \cdots, k_l + k_r\}$.

(That is, we have $k_l + k_r$ copies of each form L_i and $k_l + k_r + 1$ copies of each form R_i). To each L_{ij}, R_{ij} we assign the label 0 if $j < k_l$, and 1 if $j \geq k_l$. Finally, we set $k = (k_l + k_r - 1)(k_l + k_r + 1) + 1$.

Now, observe that the inequality (3) holds if and only if, among the $k_l + k_r - 1$ smallest forms, there are at least k_l forms L_i. Assume that there are precisely a forms L_i and b forms R_i. In particular, $a + b = k_l + k_r - 1$. Then, in the new representation, these linear forms give us

$$a(k_l + k_r) + b(k_l + k_r + 1) = (a + b)(k_l + k_r + 1) - a = (k_l + k_r - 1)(k_l + k_r + 1) - a$$

smallest forms. By construction, the next smallest forms are either $L_{i0} \leq \cdots \leq L_{i(k_l+k_r)}$ or $R_{i0} \leq \cdots \leq R_{i(k_l+k_r+1)}$ for some i. Thus, the k'th smallest form is either L_{ia} or R_{ia} and its label is 1 if and only if $a \geq k_l$ as desired. ◀

A.8 Proof of Theorem 27

Proof. Suppose we are given a function $f \in \mathsf{SYM} \circ \mathsf{MAJ}$ and a circuit computing it. We are going to construct a $\widehat{\mathsf{kSTAT}}$ representation of f in the form given by Theorem 26.

We can assume that all MAJ gates in the circuit have the same threshold $t = 0$. For this we can just add dummy variables and fix them to constants. Denote the linear forms for MAJ gates by $L_1, \cdots, L_s$ (all weights are integers) and denote by $g\colon \{0,1\}^s \to \{0,1\}$ the symmetric function at the top of the circuit. Here, s is the size of the circuit. Now, construct a $\widehat{\mathsf{kSTAT}}$ representation with the following linear forms:

$$(s+2)L_1(\boldsymbol{x}), \cdots, (s+2)L_s(\boldsymbol{x}), 1, 2, \cdots, s+1. \tag{6}$$

That is, we multiply each linear form by $(s + 2)$ and add $(s + 1)$ constant linear forms with values $1, \cdots, s + 1$. We let $k = s + 1$.

It is easy to see that the k'th statistic of (6) is always one of the constant linear forms. It is the form i if and only if $i - 1$ of the linear forms among $L_1, \cdots, L_s$ are positive. We assign label 1 to the form i if and only if $g(\boldsymbol{x}) = 1$ for inputs of weight $i - 1$. As a result, we get the desired representation for f and show that $f \in \widehat{\mathsf{kSTAT}}$. ◀

▶ Remark 46. The well-known argument that shows $\mathsf{MAJ}\circ\mathsf{THR} = \mathsf{MAJ}\circ\mathsf{MAJ}$ (see [14]) can be straightforwardly adapted to show that $\mathsf{SYM}\circ\mathsf{THR} = \mathsf{SYM}\circ\mathsf{MAJ}$. Thus, $\mathsf{SYM}\circ\mathsf{THR} \subseteq \widehat{\mathsf{kSTAT}}$ follows from Theorem 27 as well.

A.9 Proof of Theorem 28

Proof. First, as a warm-up, we show that $\mathsf{IP} \in \mathsf{kNN}$. Recall that $\mathsf{IP}(\boldsymbol{x}, \boldsymbol{y}) = \bigoplus_{i=1}^{n}(x_i \wedge y_i)$. Denote by $\boldsymbol{a} = (\frac{1}{2}, \cdots, \frac{1}{2})$ an $2n$-dimensional vector with $\frac{1}{2}$ in each coordinate. Note that $\Delta(\boldsymbol{a}, (\boldsymbol{x}, \boldsymbol{y})) = \frac{n}{2}$ for all $(\boldsymbol{x}, \boldsymbol{y}) \in \{0,1\}^{2n}$.

For each $i = 1, \ldots, n$ introduce two anchors $\boldsymbol{p_{i0}} = \boldsymbol{a} + \frac{1}{2}(\boldsymbol{e_i} + \boldsymbol{e_{i+n}})$ and $\boldsymbol{p_{i1}} = \boldsymbol{a} + \frac{1}{4}(\boldsymbol{e_i} + \boldsymbol{e_{i+n}})$. If for some $(\boldsymbol{x}, \boldsymbol{y})$ we have $x_i = y_i = 1$, then

$$\Delta((\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{p_{ij}}) \leq \frac{n}{2} - 2\left(\frac{1}{4} - \frac{1}{16}\right) = \frac{n}{2} - \frac{3}{8}.$$

If, on the other hand, $x_i = 0$ or $y_i = 0$, then

$$\Delta((\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{p_{ij}}) \geq \frac{n}{2} - \left(\frac{1}{4} - \frac{1}{16}\right) - \left(\frac{1}{4} - \frac{9}{16}\right) = \frac{n}{2} + \frac{1}{8} > \frac{n}{2}.$$

For each $i = 1, \cdots, n+1$ and $j = 0, 1$ and $l = 0, 1$ introduce an anchor $\boldsymbol{q_{i,j,l}} = \boldsymbol{a} + (-1)^l \frac{2i+j}{8n} \boldsymbol{e_1}$. For $(\boldsymbol{x}, \boldsymbol{y})$ with $x_1 = 1$ it is not hard to see that

$$\frac{n}{2} - \frac{3}{8} < \Delta((\boldsymbol{x},\boldsymbol{y}), \boldsymbol{q_{n+1,1,0}}) < \Delta((\boldsymbol{x},\boldsymbol{y}), \boldsymbol{q_{n+1,0,0}}) < \cdots < \\ < \Delta((\boldsymbol{x},\boldsymbol{y}), \boldsymbol{q_{1,1,0}}) < \Delta((\boldsymbol{x},\boldsymbol{y}), \boldsymbol{q_{1,0,0}}) < \frac{n}{2}$$

and $\Delta((\boldsymbol{x},\boldsymbol{y}), \boldsymbol{q_{i,j,1}}) > \frac{n}{2}$ for all i, j. The situation is symmetric for $x_1 = 0$. We assign label j to the anchor $\boldsymbol{p_{ij}}$. We assign label 1 to the anchor $\boldsymbol{q_{ijl}}$ iff $i + j$ is odd. We let $k = 2n + 1$.

It is easy to see that for a given $(\boldsymbol{x}, \boldsymbol{y})$ among the k closest anchors we have all pairs of anchors $\boldsymbol{p_{i0}}, \boldsymbol{p_{i1}}$ for all i such that $x_i = y_i = 1$. Denote the number of such i by t. Also among the k closest anchors we will have pairs of anchors $\boldsymbol{q_{i,0,l}}, \boldsymbol{q_{i,1,l}}$ for an appropriate l and for $i = n + 1, \ldots, t + 2$. In each of these pairs the labels of anchors are opposite and they cancel out when we compute the majority. Finally, one last anchor we will have among the k closest anchors is $\boldsymbol{q_{t+1,1,l}}$. The label of this anchor determines the majority among the k closest anchors and it is 1 iff t is odd. As a result, we get the desired representation for IP with $6n + 4$ anchors.

Now we extend this argument to SYM $\circ$ AND. Consider a function $f(\boldsymbol{x}) = g(f_1(\boldsymbol{x}), \cdots, f_s(\boldsymbol{x}))$, where each f_i has the form $f_i(\boldsymbol{x}) = \left(\bigwedge_{i \in S_i} x_i\right) \wedge \left(\bigwedge_{i \in T_i} \neg x_i\right)$ for some disjoint $S_i, T_i \subseteq [n]$. For each f_i we let $\frac{1}{2} > \epsilon_{i1} > \epsilon_{i0} > 0$ be a couple of parameters to be fixed later. We introduce a pair of anchors $\boldsymbol{p_{i1}}, \boldsymbol{p_{i0}}$ in the following way: Set the kth coordinate of $\boldsymbol{p_{ij}}$ to

$$\boldsymbol{p_{ij}}^{(k)} = \begin{cases} 1/2 & k \notin S_i \cup T_i \\ 3/2 - \epsilon_{ij} & k \in S_i \\ \epsilon_{ij} - 1/2 & k \in T_i \end{cases}$$

It is easy to see that for $\boldsymbol{x}$ such that $f_i(\boldsymbol{x}) = 1$ we have $\Delta(\boldsymbol{p_{ij}}, \boldsymbol{x}) = \frac{n}{4} - |S_i \cup T_i|(\epsilon_{ij} - \epsilon_{ij}^2)$ and for $\boldsymbol{x}$ such that $f_i(\boldsymbol{x}) = 0$ we have $\Delta(\boldsymbol{p_{ij}}, \boldsymbol{x}) \geq \frac{n}{4} - (|S_i \cup T_i| - 1)(\epsilon_{ij} - \epsilon_{ij}^2) + 1$. We fix ϵ_{ij} in such a way that $\frac{n}{4} - |S_i \cup T_i|(\epsilon_{ij} - \epsilon_{ij}^2) < \frac{n}{4} - \frac{1}{2}$ and $\frac{n}{4} - (|S_i \cup T_i| - 1)(\epsilon_{ij} - \epsilon_{ij}^2) + 1 > \frac{n}{4} + \frac{1}{2}$. We set $\mathsf{label}(\boldsymbol{p_{ij}}) = j$.

We construct anchors $\boldsymbol{q_{ijl}}$ for $i = 1, \cdots, s + 1$ and $j = 0, 1$ the same way as above and assign $\mathsf{label}(\boldsymbol{q_{i1l}})$ to be equal to $g(\boldsymbol{y})$ for $\boldsymbol{y}$ of weight $i - 1$ and $\mathsf{label}(\boldsymbol{q_{i0l}})$ to be the opposite. We let $k = 2s + 1$. The same argument as for IP shows that we get the desired representation of f with $6s + 4$ anchors. ◀

A.10 Proof of Theorem 30

Proof. Consider a function $f \in$ ELDL and suppose the linear forms in its representation are $L_1, \cdots, L_s$. Here L_i corresponds to the i'th query. As in the proof of Theorem 27, we can assume that all thresholds in all linear forms are 0.

We are going to construct a representation for f of the form provided by Theorem 26. We add to this representation the following linear forms:

$$(s+1)L_1, -(s+1)L_1, (s+1)L_2+1, -(s+1)L_2-1, \cdots, (s+1)L_s+s-1, -(s+1)L_s-s+1.$$

That is, for each form L_i in ELDL representation, we add the two forms $(s + 1)L_i + (i - 1)$ and $-(s + 1)L_i - (i - 1)$. We set $k = s$.

Assume that for some x we have $L_i(\boldsymbol{x}) = 0$ and all previous linear forms are non-zero. We than have that $(s+1)L_i(\boldsymbol{x}) + i - 1 = i - 1$. It is not hard to see that for $j < i$ we have that among forms $(s+1)L_j(\boldsymbol{x}) + j - 1$ and $-(s+1)L_j(\boldsymbol{x}) - j + 1$ exactly one is greater than $i - 1$: it is the first one if $L_j(\boldsymbol{x}) > 0$ and the second one if $L_j(\boldsymbol{x}) < 0$. For $j > i$ in a similar way we can see that among the forms $(s+1)L_j(\boldsymbol{x}) + j - 1$ and $-(s+1)L_j(\boldsymbol{x}) - j + 1$ exactly one is greater than $i - 1$: it is the first one if $L_j(\boldsymbol{x}) \geq 0$ and the second one if $L_j(\boldsymbol{x}) < 0$. As a result there are exactly $s - 1$ forms that are greater than $(s+1)L_i(\boldsymbol{x}) + i - 1$. We assign to this form the same label $L_i(\boldsymbol{x})$ has in ELDL. From this it follows that the constructed representation computes the same function.

Clearly, the coefficients in the constructed form are polynomially related to the coefficients in the original forms. Thus, the same proof gives $\widehat{\mathsf{ELDL}} \subseteq \widehat{\mathsf{kSTAT}}$. ◀

▶ Remark 47. Note that decision lists are computable in AC^0 and thus can be computed by quasi-polynomial-size $\mathsf{SYM} \circ \mathsf{AND}$ circuits. As a result, ELDL can be computed by quasi-polynomial-size circuit in $\mathsf{SYM} \circ \mathsf{AND} \circ \mathsf{ETHR} = \mathsf{SYM} \circ \mathsf{ETHR} = \mathsf{SYM} \circ \mathsf{MAJ}$, where the second equality follows since ETHR is closed under AND operation. Still, Theorem 30 gives a polynomial reduction that translates to the case of small coefficients.

A.11 Proof of Theorem 37

Such constructions are likely known; we outline a simple one for completeness.

▶ **Lemma 48.** *For any even integer $k > 0$, there exists a CNF with n variables and $n2^{k-o_k(k)}/k$ clauses with $2^{(1-o_k(1))n}$ components.*

Proof. Assume k divides n. Divide the set of variables to n/k disjoint sets $S_1, \cdots, S_{n/k}$ of size k. For each set S_i, define a CNF C_i which evaluates to 1 if and only if exactly half of the variables in S are equal to 1. This can be achieved with $\binom{k}{k/2} = 2^{k-o_k(k)}$ clauses.

Then, the CNF $C = C_1 \wedge \cdots \wedge C_{n/k}$ has exactly $\binom{k}{k/2}^{n/k} = 2^{(1-o_k(1))n}$ satisfying assignments, and the Hamming distance between any two of such assignments is at least 2. Thus, each of them constitutes a component. ◀

Hence, Theorem 37 follows from Lemmas 36 and 48 by taking k to be a constant independent of n. It is easy to extend the construction above to odd k. We omit the simple details.

B Circuits computing nearest neighbors

In this section we describe a straightforward construction of a *depth-three* circuit computing HNN and then compress it to *depth-two* at the cost of exponential weights. The folklore result of [31] is that any NN representation with m anchors can be computed by a depth three threshold circuit with size $O(m^2)$. A short proof can be found in [24].

▶ **Theorem 49** ([31])**.**
- $\mathsf{NN} \subseteq \mathsf{OR} \circ \mathsf{AND} \circ \mathsf{THR}$, $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{THR}$
- $\mathsf{HNN} \subseteq \mathsf{OR} \circ \mathsf{AND} \circ \mathsf{MAJ}$, $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{MAJ}$

Namely, every NN *(*HNN*) representation is computed by a depth-three* $\mathsf{AC}^0 \circ \mathsf{THR}$ *(*MAJ*) circuit with size* $|P||N| + \min\{|P|, |N|\} + 1$.

Note that the only difference between the circuits for HNN and NN is that the first-level threshold gates are guaranteed to have polynomial weights (in the case of HNN). It turns out that the size of the HNN circuit can be improved (when $n \ll |P| + |N|$).

▶ **Lemma 50.**

$$\mathsf{HNN} \subseteq \mathsf{OR} \circ \mathsf{AND} \circ \mathsf{MAJ}$$

In particular, every HNN *representation with m anchors is computed by an* $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{THR}$ *circuit with size $(n+1)m + (n+1)|P| + 1$.*

Proof. Note that $\mathbb{1}[\Delta(\boldsymbol{x}, \boldsymbol{p}) \leq i]$ is computed by a threshold gate $f^{\boldsymbol{p}}_{\leq i}(\boldsymbol{x})$ defined by $\boldsymbol{w} = \boldsymbol{p} - \overline{\boldsymbol{p}}$ and $\theta = \Delta(\boldsymbol{p}) - i$. (And similarly $\mathbb{1}[\Delta(\boldsymbol{x}, \boldsymbol{p}) \geq i]$.) Suppose f has an HNN representation $P \cup N$. Then, $f(\boldsymbol{x}) = \bigvee_{\substack{i \leq n \\ \boldsymbol{p} \in P}} \left(f^{\boldsymbol{p}}_{\leq i}(\boldsymbol{x}) \wedge \bigwedge_{\boldsymbol{q} \in N} f^{\boldsymbol{q}}_{\geq i}(\boldsymbol{x}) \right)$ ◀

Note that the threshold circuits from Theorem 49 and Lemma 50 have size $O(m^2)$ and $O(mn)$ respectively. In fact, the latter circuit can be compressed to a depth-two threshold circuit with exponential weights.

▶ **Theorem 51.**

$$\mathsf{HNN} \subseteq \mathsf{THR} \circ \mathsf{MAJ}.$$

Namely, every HNN *representation with m anchors is computed by a threshold of $2nm$ majority gates.*

Proof. The first level will consist of $2mn$ gates $f^{\boldsymbol{p}}_{\leq i}$, $f^{\boldsymbol{p}}_{\geq i}$ which output 1 if and only if $\Delta(\boldsymbol{x}, \boldsymbol{p}) \leq i$ and $\Delta(\boldsymbol{x}, \boldsymbol{p}) \geq i$, respectively, for $1 \leq i \leq n$. Define the sum

$$g^p_i(\boldsymbol{x}) := f^{\boldsymbol{p}}_{\leq i}(\boldsymbol{x}) + f^{\boldsymbol{p}}_{\geq i}(\boldsymbol{x}) - 1$$

and note that $g^p_i(\boldsymbol{x}) = \mathbb{1}[\Delta(x, p) = i]$. We can then write the output gate as

$$h(\boldsymbol{x}) = \mathbb{1}\left(\sum_{p \in P, i \leq n} m^{3(n-i)+1} g^p_i(\boldsymbol{x}) - \sum_{q \in N, i \leq n} m^{3(n-i)} g^q_i(\boldsymbol{x}) \geq 0 \right).$$

If some positive anchor is at distance at most j and all negative anchors are at distance at least j to $\boldsymbol{x}$, then

$$\sum_{p \in P, i \leq n} m^{3(n-i)+1} g^p_i(\boldsymbol{x}) \geq m^{3(n-j)+1} \geq \sum_{q \in N, i \leq n} m^{3(n-i)} g^q_i(\boldsymbol{x}).$$

Conversely, if some negative anchor is at distance at most j and all positive anchors are at distance at least $j+1$, then

$$\sum_{p \in P, i \leq n} m^{3(n-i)+1} g^p_i(\boldsymbol{x}) \leq m^{3(n-j)-1} < m^{3(n-j)} \leq \sum_{q \in N, i \leq n} m^{3(n-i)} g^q_i(\boldsymbol{x}).$$ ◀

▶ Remark 52. Theorem 51 can be obtained through Theorem 14, as a consequence of the following result derived from [18]. We include the direct construction to avoid the slight increase in circuit size.

▶ **Lemma 53.**

$$\mathsf{mpPTF}(\mathrm{poly}(n)) \subseteq \mathsf{THR} \circ \mathsf{MAJ}$$

Every mpPTF *with ℓ terms and maximum weight W is computed by a linear threshold of at most $4 \cdot W\ell \log \ell$ majority gates.*

Proof. Let $\mathsf{PTF}_{1,2}$ refer to Boolean functions (over $\{1,2\}$) equal to the sign of an n-variate polynomial. [18] prove that any $\mathsf{PTF}_{1,2}$ with ℓ terms and degree at most d is computed by a linear threshold (with exponential weights) of at most $2\ell d$ majority gates (replacing $\{1,2\}$ with $\{0,1\}$), and any mpPTF with ℓ terms and maximum weight W can be represented by a $\mathsf{PTF}_{1,2}$ with ℓ terms and degree at most $2W \log \ell$. ◀

▶ Remark 54. It is not hard to see that the circuits constructed in this section are *polynomial-time uniform*; they can be generated by a Turing machine given the set of anchors in polynomial time.

Space Complexity of Minimum Cut Problems in Single-Pass Streams

Matthew Ding
University of California, Berkeley, CA, USA

Alexandro Garces
Massachusetts Institute of Technology, Cambridge, MA, USA

Jason Li
Carnegie Mellon University, Pittsburgh, PA, USA

Honghao Lin
Carnegie Mellon University, Pittsburgh, PA, USA

Jelani Nelson
University of California, Berkeley, CA, USA

Vihan Shah
University of Waterloo, Canada

David P. Woodruff
Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

We consider the problem of finding a minimum cut of a weighted graph presented as a single-pass stream. While graph sparsification in streams has been intensively studied, the specific application of finding minimum cuts in streams is less well-studied. To this end, we show upper and lower bounds on minimum cut problems in insertion-only streams for a variety of settings, including for both randomized and deterministic algorithms, for both arbitrary and random order streams, and for both approximate and exact algorithms. One of our main results is an $\widetilde{O}(n/\varepsilon)$ space algorithm with fast update time for approximating a spectral cut query with high probability on a stream given in an arbitrary order. Our result breaks the $\Omega(n/\varepsilon^2)$ space lower bound required of a sparsifier that approximates all cuts simultaneously. Using this result, we provide streaming algorithms with near optimal space of $\widetilde{O}(n/\varepsilon)$ for minimum cut and approximate all-pairs effective resistances, with matching space lower-bounds. The amortized update time of our algorithms is $\widetilde{O}(1)$, provided that the number of edges in the input graph is at least $(n/\varepsilon^2)^{1+o(1)}$. We also give a generic way of incorporating sketching into a recursive contraction algorithm to improve the post-processing time of our algorithms. In addition to these results, we give a random-order streaming algorithm that computes the *exact* minimum cut on a simple, unweighted graph using $\widetilde{O}(n)$ space. Finally, we give an $\Omega(n/\varepsilon^2)$ space lower bound for deterministic minimum cut algorithms which matches the best-known upper bound up to polylogarithmic factors.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases minimum cut, approximate, random order, lower bound

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.43

Related Version *Full Version*: https://arxiv.org/abs/2412.01143 [17]

Funding *Matthew Ding*: Supported in part by NSF CCF-1951384.
Alexandro Garces: Supported in part by NSF CNS-2150186.
Jason Li: This work was done in part as a Research Fellow at the Simons Institute for the Theory of Computing.
Honghao Lin: Supported in part by a Simons Investigator Award, NSF CCF-2335412, and a CMU Paul and James Wang Sercomm Presidential Graduate Fellowship.



© Matthew Ding, Alexandro Garces, Jason Li, Honghao Lin, Jelani Nelson, Vihan Shah, and David P. Woodruff;
licensed under Creative Commons License CC-BY 4.0



Jelani Nelson: Supported in part by NSF CCF-1951384 and NSF CCF-2427808.
Vihan Shah: Supported in part by Sepehr Assadi's Sloan Research Fellowship and NSERC Discovery Grant.
David P. Woodruff: Supported in part by a Simons Investigator Award and NSF CCF-2335412.

Acknowledgements The authors would like to thank the ITCS 2025 reviewers for their anonymous feedback. Alexandro Garces and Vihan Shah are extremely grateful to Sepehr Assadi for many helpful conversations throughout the project. They also thank the organizers of DIMACS REU in Summer 2023, in particular Lazaros Gallos, for initiating this collaboration and for all their help and encouragement along the way.

1 Introduction

We consider the graph streaming model, which is a key model for computations on massive graph datasets that has been extensively studied over the past couple of decades (see, e.g., [36] for a survey). We specifically study the problem of finding minimum cuts in adversarial and random-order streams, which has been less studied than the problem of general cut sparsification of graphs. Besides being theoretically interesting, finding minimum cuts is also a problem of practical interest. For example, they allow for the calculation of social network metrics such as influence [50] and are used to quantify the robustness of power networks [23] and road networks [48] to failures.

1.1 Adversarial Streams

The key method so far for solving minimum cut with low-memory is the usage of cut sparsifiers. The notion of cut sparsifiers was introduced by Benczur and Karger [7] and has been extremely influential. Given a weighted graph $G = (V, E, w)$ with $n = |V|$ vertices and $m = |E|$ edges, and polynomially bounded edge weights $w : E \to \mathbb{R}_+$, together with an accuracy parameter $\varepsilon > 0$, a cut sparsifier of G is a sparse subgraph H on the same vertex set V but with (possibly) different edge weights such that the weight of every cut in G is $(1+\varepsilon)$-approximated by the weight of the corresponding cut in H. For two sets $S, T \subseteq V$, we use $E(S,T) = \{(u,v) \in E : u \in S, v \in T\}$ to denote the set of edges between S and T in graph G and $w_G(S,T) = \sum_{e \in E(S,T)} w_e$ to denote the total weight of edges between S and T in graph G. Formally, we have the following definition,

▶ **Definition 1** (For-All Cut Sparsifier). *H is a $(1+\varepsilon)$ for-all cut sparsifier of G if and only if the following holds for all $\varnothing \subset S \subset V$:*

$$w_H(S, V \setminus S) \approx_\varepsilon w_G(S, V \setminus S)$$

where $a \approx_\varepsilon b$ is defined as $(1-\varepsilon) \cdot b \leq a \leq (1+\varepsilon) \cdot b$.

Specifically, [7] shows that a cut sparsifier always exists with $O(n \log n/\varepsilon^2)$ edges. This bound was improved to $O(n/\varepsilon^2)$ edges by [6] and [1, 2, 11] proved a lower bound of $\Omega(n(\log n)/\varepsilon^2)$ bits (even in the case when the cut-sparsifier H is a not necessarily a subgraph of G). The existence bound was also extended to the stronger notion of spectral sparsifiers [46, 47, 45, 6], where the quadratic form associated with the Laplacian of graph H provides a $(1+\varepsilon)$-approximation to that of G. Let L_G denote the Laplacian matrix of graph G. For every subset $S \subseteq V$, let x_S be the binary indicator vector of S such that $x_i = 1$ if $i \in S$ and $x_i = 0$ if $i \notin S$. Then we have $x_S^\top L_G x_S = w_G(S, V \setminus S)$. The definition of the spectral sparsifier extends the assumption that x is a binary vector to an arbitrary vector.

▶ **Definition 2** (For-All Spectral Sparsifier). *Let G and H be two weighted undirected graphs. Fix $0 < \varepsilon < 1$. We say H is a $(1+\varepsilon)$ for-all spectral sparsifier of G iff the following holds:*

$$(1-\varepsilon) \cdot L_G \preceq L_H \preceq (1+\varepsilon) \cdot L_G$$

where we use $A \preceq B$ to denote $\forall x \in \mathbb{R}^n, x^\top Ax \leq x^\top Bx$.

This construction has had a tremendous impact on cut problems in graphs, see, e.g., [7, 8, 30, 44, 35]. However, for very small values of ε, the $1/\varepsilon^2$ dependence in cut sparsifiers may be prohibitive on large-scale graphs. Motivated by this, the work of [2] relaxed the cut sparsification problem to that of outputting a data structure sk_G such that given any *fixed* cut $S \subseteq V$, the value of $\mathrm{sk}_G(S)$ is within a $(1+\varepsilon)$ factor of the cut value of S in G, with probability at least $2/3$[1]. This relaxed notion of the problem is called the "for-each" model, which should be contrasted with the previous "for-all" model of cut and spectral sparsification.

Surprisingly, [2] showed that such a data structure exists for cut sparsification with $\mathrm{poly}(n)$-bounded integer edge weights of size $\widetilde{O}(n/\varepsilon)$[2], which is optimal up to polylogarithmic factors. In a follow-up work of [24], the authors extended the $\widetilde{O}(n/\varepsilon)$ upper bound to the for-each model for spectral sparsifiers. Another very interesting work is [14], which shows that such a sketch can be chosen as a reweighted subgraph of G. We specifically deal with these graphical for-each sparsifiers:

▶ **Definition 3** (Graphical For-Each Spectral Sparsifier). *Let G and H be two weighted undirected graphs. Fix $0 < \varepsilon < 1$. We say H is a $(1+\varepsilon)$ graphical for-each spectral sparsifier of G if and only if for each $x \in \mathbb{R}^n$, with probability at least $2/3$:*

$$x^\top L_H x \approx_\varepsilon x^\top L_G x$$

In a line of work, see, e.g., [36, 27, 28], efficient algorithms for sparsifiers in the for-all model in a graph stream were constructed. The state-of-the-art work [28] gives a single-pass algorithm for spectral sparsification in dynamic streams which uses space $\widetilde{O}(n/\varepsilon^2)$.

▶ **Lemma 4** ([28]). *Given an input graph G in a stream, there exists a one-pass streaming algorithm that outputs a subgraph H of G such that with high probability H is a $(1+\varepsilon)$-spectral sparsifier of G. Moreover, the algorithm uses $\widetilde{O}(n/\varepsilon^2)$ space and $\widetilde{O}(m + n/\varepsilon^2)$ time.*

However, it was not known how to construct a for-each sketch for cut sparsification in a stream in better than $\widetilde{O}(n/\varepsilon^2)$ space, which follows from computing a for-all sparsifier. A natural question is whether we can implement a streaming algorithm that computes a for-each sparsifier with space matching the $\Omega(n/\varepsilon)$ offline lower bound that holds for for-each sparsification [2].

If we can answer this question in the affirmative, we can also get a better space bound for the approximate minimum cut problem. It is well-known that there are only $\mathrm{poly}(n)$ $O(1)$-approximate minimum cuts[3] in a graph. So we can run a for-all cut sparsification algorithm in a stream with $\varepsilon = \Theta(1)$ in $\widetilde{O}(n)$ space and obtain a candidate set of $\mathrm{poly}(n)$

[1] This can be amplified to high probability, which we define to mean with probability at least $1 - 1/\mathrm{poly}(n)$. This is done by independently repeating the data structure $O(\log n)$ times and outputting the median estimate. This also lets us estimate the cut value for $\mathrm{poly}(n)$ cuts simultaneously.

[2] Throughout we use $\widetilde{O}(\cdot)$ to hide $\mathrm{polylog}(n)$ factors.

[3] When we say a cut is an α-approximate minimum cut, we mean that its cut value is at most α times the minimum cut value.

cuts containing the minimum cut. In parallel, we estimate the value of each cut using a for-each sparsifier with failure probability $1/\text{poly}(n)$. The smallest cut out of the candidate cuts will be a $(1+\varepsilon)$-approximate minimum cut. Thus, the space complexity is just $\widetilde{O}(n)$ plus that of maintaining a for-each sparsifier in a stream. However, the existing algorithms for computing for-each sparsifiers are not streaming algorithms. Motivated by the existence of a for-each sketch in $\widetilde{O}(n/\varepsilon)$ bits of space, one may wonder if it is possible to obtain a $(1+\varepsilon)$-approximation to the minimum cut in $\widetilde{O}(n/\varepsilon)$ space in a single-pass stream.

▶ **Question 5.** What is the space and time complexity of obtaining a $(1+\varepsilon)$-approximate minimum cut in a single-pass insertion-only stream?

We also note that an exact global minimum cut streaming algorithm exists in $\widetilde{O}(n)$ space with two passes for simple graphs [5] (see also [43]) and $\log n$ passes for weighted graphs [38].

1.2 Random-order streams

We know that finding the minimum cut exactly in single-pass adversarial streams needs $\Omega(n^2)$ space [52]. We need an additional pass to get the exact minimum cut in $\widetilde{O}(n)$ space. However, many streaming applications require solving the problem in exactly one pass because the stream cannot be stored.

Thus, to surpass the $\Omega(n^2)$ barrier of adversarial streams in one pass, we consider a relaxation known as random-order streams and ask whether we can find the minimum cut exactly in $o(n^2)$ space in this setting. In this model, the graph can still be chosen adversarially, but its edges arrive in the stream in a random order.

The motivation for studying this model stems from the fact that real-world data is typically not adversarial. Adversarial streams are adversarial in both their input and arrival order, while random-order streams relax the arrival order, making them more representative of real-world data. Additionally, the random-order streaming model may help explain the empirical performance of certain heuristics.

This model was first studied by [39] for sorting in limited space. Many subsequent works have studied problems in random-order streams [21, 26, 12, 37, 42, 16, 13], some of which show a clear separation between the adversarial and random-order models.

One notable line of work focuses on maximum matching within this model [33, 19, 32, 3, 18, 9, 4, 22]. The best-known upper bound in the random order model [4] gets a slightly better than 2/3-approximation in semi-streaming space, which is better than the best possible $\frac{1}{1+\ln 2} \approx 0.59$-approximation in the adversarial model due to the lower bound of [25], showing another separation between adversarial and random-order streams.

Motivated by this, we study the minimum cut problem in the random-order model and ask the following question:

▶ **Question 6.** What is the space complexity of obtaining the exact minimum cut in a single-pass random-order stream?

This work resolves the above questions with matching algorithms and lower bounds. We also carefully optimize the running time and update time of our algorithms. We also obtain several additional results, which we outline in Section 3.

2 Preliminaries

2.1 All-Pairs Effective Resistances

Given $G(V, E, w)$ as an electrical network on n nodes in which each edge e corresponds to a link of conductance w_e (i.e., a resistor of resistance $1/w_e$), the effective resistance of an edge e is the potential difference induced across it when a unit current is injected at one end of e and extracted at the other end of e. Equivalently it is equal to $R^G_{\text{eff}} = \chi^\top_{u,v} L^\dagger_G \chi_{u,v}$, where $L^\dagger_G$ denotes the pseudoinverse of the Laplacian of graph G and $\chi_{u,v} := 1_u - 1_v$.

In the all-pairs effective resistance problem, we are required to output a data structure, which with high probability can generate effective resistance values for each of the n^2 pairs of vertices, or a $(1+\varepsilon)$-approximation of each of the n^2 pairs of values in the $(1+\varepsilon)$-approximate version of the problem. Note that the data structure implicitly represents these n^2 values and responds with one such value on a given query pair.

2.2 Short-Cycle Decomposition

Short cycle decompositions are a recent algorithmic tool for graph sketching problems introduced by Chu et al. [14].

▶ **Definition 7** (Short-Cycle Decomposition). *An $(\hat{m}, L)$ short-cycle decomposition of a graph G is a decomposition of the graph into edge-disjoint cycles of at most length L and an additional $\hat{m}$ edges outside of these cycles.*

The main contribution of [14] to spectral sparsification is in proving the following claim:

▷ Claim 8. Given an undirected weighted graph G and an $(\hat{m}, L)$ short-cycle decomposition routine CycleDecomp, there exists an algorithm which returns with high probability a $(1+\varepsilon)$-spectral sparsifier *graph* with $\widetilde{O}(\hat{m} + nL\varepsilon^{-1})$ edges and run-time

$$\widetilde{O}(m) + T_{\text{CycleDecomp}}(O(m \log n), n)$$

where $T_{\text{CycleDecomp}}(m, n)$ is the running time of CycleDecomp on a graph with m edges and n vertices.

As a high-level overview, the short-cycle decomposition is key as a degree-preserving sparsification method. Given each cycle, we can label them numerically in order and sample either all the odd or all the even labeled edges, each with $1/2$ probability. Previous works on "for-each" sparsifiers [2, 24] used a recursive expander decomposition and subsampling at each level, with the main issue being that the degrees were not well-preserved. This issue was solved by explicitly storing the degrees of the graph, causing their spectral sparsifier data structure to no longer be a graph. The work of [14] circumvents this issue with the short-cycle decomposition by constructing subsampled graphs that exactly preserve their vertex degree.

Note that [14] already gives a graphical spectral sparsifier with $\widetilde{O}(n/\varepsilon)$ edges and bits of working memory using a basic brute-force search for cycle decomposition. However, its runtime for construction is $O(mn)$, which is prohibitively slow as a subroutine. On the other hand, later work by Parter and Yogev [41] shows a deterministic algorithm to compute $(O(n \log n), O(\log^2 n))$ short-cycle decompositions. This combined with Claim 8 will allow us to design an algorithm in $m^{1+o(1)}$ time which returns a $(1+\varepsilon)$-spectral sparsifier which is a reweighted graph with $\widetilde{O}(n/\varepsilon)$ edges. This gives near-optimal parameters except for a required $m^{1+o(1)}$ bits of working memory. However, in Section 4.1, we show that we can implement this subroutine so that it is guaranteed to use only $\widetilde{O}(m)$ working memory, achieving near-optimal results in both working memory usage and running time.

Table 1 Minimum Cut Space Complexity in the Single-Pass Insertion-Only Streaming Setting.

Stream Type	Exact/Approx $(1+\varepsilon)$	Upper Bound	Lower Bound
Adversarial	Exact	$O(n^2)$ (full graph)	$\Omega(n^2)$ [52]
Adversarial	Approx, Deterministic	$\widetilde{O}(n/\varepsilon^2)$ [6]	$\Omega(n/\varepsilon^2)^*$ (Theorem 12)
Adversarial	Approx, Randomized	$\widetilde{O}(n/\varepsilon)$ (Theorem 9)	$\Omega(n/\varepsilon)$ (Theorem 12)
Random-Order	Exact	$\widetilde{O}(n)$ (Theorem 14)	$\Omega(n)$ [12]

3 Overview of Results

We list the current state-of-the-art results for single-pass minimum cut streaming algorithms in Table 1. Our main result is an $\widetilde{O}(n/\varepsilon)$ space streaming algorithm that finds a $(1+\varepsilon)$-approximate minimum cut in a single-pass stream.

▶ **Theorem 9.** *There is a one-pass insertion-only streaming algorithm that, with high probability, computes an $(1+\varepsilon)$-approximation of the minimum cut on weighted graphs using $\widetilde{O}(n/\varepsilon)$ bits of space. Moreover, our algorithm takes $\widetilde{O}(m)+(n/\varepsilon^2)^{1+o(1)}$ total update time and $\widetilde{O}(n^2/\varepsilon^2)$ post-processing time.*

This shows, somewhat surprisingly, that estimating the minimum cut is easier than computing a for-all cut sparsifier in a data stream. Our algorithm is mainly based on the following new for-each spectral sparsifier in a graph stream.

▶ **Lemma 10.** *A one-pass insertion-only streaming algorithm exists that, with high probability, constructs a $(1+\varepsilon)$ for-each spectral sparsifier of weighted graphs with $\widetilde{O}(n/\varepsilon)$ edges using $\widetilde{O}(n/\varepsilon)$ bits of space. This algorithm also has total runtime $\widetilde{O}(m)+(n/\varepsilon^2)^{1+o(1)}$.*

Our algorithm's total runtime is $\widetilde{O}(m)+(n/\varepsilon^2)^{1+o(1)}$, which notably implies its amortized update time is $\widetilde{O}(1)$ when $m \geq (n/\varepsilon^2)^{1+o(1)}$. With the $\Omega(n/\varepsilon)$ data structure lower bound in [2], our algorithm is tight in space complexity up to polylogarithmic factors. Based on our Lemma 10, we can obtain an $\widetilde{O}(n/\varepsilon)$ space streaming algorithm that can $(1\pm\varepsilon)$-estimate all-pairs effective resistances.

▶ **Corollary 11.** *There exists a one-pass insertion-only streaming algorithm that constructs a data structure that calculates $(1+\varepsilon)$-approximations to the effective resistances between every pair of vertices on weighted graphs using $\widetilde{O}(n/\varepsilon)$ bits of space. Our algorithm has $\widetilde{O}(m)+(n/\varepsilon^2)^{1+o(1)}$ total time during the stream and $\widetilde{O}(n^2/\varepsilon)$ post-processing time.*

Notice that we are able to match the sketching bounds of [24] up to polylogarithmic factors in the space and time required.

We also show an $\Omega(n/\varepsilon)$ lower bound for minimum cut and all-pairs effective resistances in Section 4.4 and Section 4.6 respectively, showing that our results of Theorem 9 and Corollary 11 are space-optimal up to polylogarithmic factors.

▶ **Theorem 12.** *Fix $\varepsilon > 1/n$. Any randomized algorithm that outputs a $(1+\varepsilon)$-approximation to the minimum cut of a simple, undirected graph in a single pass over a stream with probability at least $2/3$ requires $\Omega(n/\varepsilon)$ bits of space. If the algorithm is deterministic and $\varepsilon \geq 1/n^{1/4}$, then the algorithm requires $\Omega(n/\varepsilon^2)$ bits of space.*[4]

[4] This latter assumption on ε is the same as used in [11].

▶ **Theorem 13.** *Fix $\varepsilon > 1/n$. Suppose $sk(\cdot)$ is a sketching algorithm that outputs at most $s = s(n, \varepsilon)$ bits, and f is an estimation algorithm such that,*

$$\forall a, b \in V, \qquad \Pr[f(a, b, sk(G)) \in (1 \pm \varepsilon) r_{a,b}] \geq \frac{2}{3},$$

where $r_{a,b}$ is the effective resistance of nodes a, b. Then we have $s \geq \Omega(n/\varepsilon)$.

We also study the minimum cut problem in the random-order streaming model, where edges arrive in a random order instead of an arbitrary worst-case order. We prove the following result for *simple* unweighted graphs, which are graphs with at most one edge between any two vertices.

▶ **Theorem 14.** *There exists a one-pass insertion-only streaming algorithm in the random-order model that outputs all minimum cuts $(S, V \setminus S)$, along with their corresponding edges $E(S, V \setminus S)$, for a simple, unweighted graph and with high probability using $\widetilde{O}(n)$ space. The algorithm has $\widetilde{O}(n)$ update time and $\widetilde{O}(n^2 k)$ post-processing time, where k is the value of the minimum cut.*

We note that Theorem 14 provides the exact minimum cut, which we find surprisingly possible in a single pass. This further adds to the surprises regarding minimum cut in the streaming model, as a 2-pass exact algorithm in arbitrary order streams was known [5] while computing the exact minimum cut in a single pass in an arbitrary order stream is known to require $\Omega(n^2)$ memory [52].

Given that determining whether a graph is connected from a random order stream requires $\Omega(n)$ space [12], our $\widetilde{O}(n)$ space algorithm for finding the exact minimum cut in random-order streams is optimal up to polylogarithmic space factors, even for just outputting the value. Note that we compute the minimum cut and all edges crossing the cut in addition to the value.

3.1 Our Techniques

Approximate Minimum Cut

Suppose that H is a $(1+\varepsilon)$-for-all sparsifier of G. Then the minimum cut of H is a $(1+\varepsilon)$-approximation to that of G. However, such a conclusion will not hold if H is instead a $(1+\varepsilon)$ for-each sparsifier, as the total number of cuts is exponential. As discussed earlier, fortunately the number of 1.1-approximate minimum cuts is $O(n^2)$ [29]. Hence, if we run a for-all sparsifier algorithm in parallel with accuracy $\varepsilon' = O(1)$, we can obtain a list of $O(n^2)$ candidate minimum cuts in G. Using the the $(1+\varepsilon)$ for-each sparsifier, we can then find a $(1+\varepsilon)$-approximation to the minimum cut by union bounding over the $O(n^2)$ candidates.

For the lower bound, we consider the k-edge-connectivity problem. The *edge-connectivity* of a graph is the minimum number of edges that need to be deleted to disconnect the graph. The k-edge-connectivity problem asks whether the edge connectivity of a graph is $< k$ or $\geq k$. Suppose that there is an algorithm that solves the $(1+\varepsilon)$-approximate minimum cut value problem. Then, we can use it to solve the k-edge-connectivity problem with $k = O(1/\varepsilon)$. Combining with the $\Omega(kn)$ bit space lower bound in the work of [49], it follows that there is an $\Omega(n/\varepsilon)$ bit space lower bound.

For-Each Spectral Sparsifier

The main difficulty in implementing the algorithms of [2] and [24] in the streaming setting is that both algorithms require careful graph decomposition. Take the algorithm in [2] as an example: it partitions the graph into several components, where each component is

well-connected and does not have a sparse cut smaller than $1/\varepsilon$. In the worst case, such a procedure may take $\log n$ levels recursively, which seems unachievable in a single-pass stream. Besides, since the for-each sketch of both algorithms is not a graph, it is unclear how to merge two sketches directly. Indeed, natural ways of merging such sketches may destroy the decomposition into sparse cuts. To address these issues, we instead consider the work of [14] in which the authors show how to construct a for-each sparsifier which is, in fact, a re-weighted subgraph of the input graph, with $n^{1+o(1)}/\varepsilon$ edges. Combining this and the merge-and-reduce framework gives a for-each spectral sparsifier with $n^{1+o(1)}/\varepsilon$ edges in a single-pass stream. The extra $n^{o(1)}$ factor in the space is an issue here. We notice that the extra $n^{o(1)}$ factor in the work of [14] is due to their near-linear time short cycle decomposition algorithm. Leveraging ideas from [41], we give a new short-cycle decomposition algorithm that trades off running time for space, namely, it achieves $m^{1+o(1)}$ time and $\widetilde{O}(m)$ space. While this translates to $\widetilde{O}(n/\varepsilon)$ space, it unfortunately gives $n^{o(1)}$ update time per edge. To fix this, we use online leverage score sampling to produce a virtual stream of only $\widetilde{O}(n/\varepsilon^2)$ edges that we instead run our algorithm on. By doing this, we reduce the time to $\widetilde{O}(1)$ per edge provided the number m of edges in the input satisfies $m \geq (n/\varepsilon^2)^{1+o(1)}$.

Approximate All-Pairs Effective Resistance

In [24], the authors show if we can get a for-each sparsifier, then we can use it to generate a data structure in near-linear time, and such a data structure can approximate all pairs of the effective resistance in $\widetilde{O}(n^2/\varepsilon)$ time. Combining this and our Lemma 10 yields our upper bound.

For the lower bound, we use communication complexity and consider a similar graph construction in [2], though we use new arguments based on random walks. Specifically, given a random binary string $s \in \{0,1\}^{n/\varepsilon}$, we encode it into a graph G where G is divided into $O(\varepsilon n)$ disjoint bipartite graphs G_i, and in each G_i, the existence of an edge between each pair corresponds to one random bit in s. We show that for every such pair (a, b), with high probability there is a $(1+\varepsilon)$-separation between the two cases (there is an edge between a, b or not), which yields an $\Omega(n/\varepsilon)$ lower bound. To prove this, in particular, we use the connection between the effective resistance and the hitting time on a graph and a recent concentration result about the hitting time on a random graph $G(n, p)$ (each possible edge on n vertices is included independently with probability p).

Exact Minimum Cut in Random-order Streams

Recall that we assume that the graph is simple and unweighted, and each edge arrives in a random order stream. We also assume the minimum cut size is $\Omega(\log n)$. If the size is smaller, a for-all cut sparsifier with accuracy parameter $\varepsilon' = 1/\log^2 n$ gives the exact value of the minimum cut. The main idea here is that by looking at the prefix of edges in a stream, which we will call the graph H, we roughly learn the sizes of all the cuts up to a small constant factor. Using this information, we learn all 1.1-approximate minimum cuts in G. The next question is how to get the exact value of all these cuts (so we can find the minimum one). For the cut edges in H, note that at this time, the cut values of all these cuts in H is $\Theta(\log n)$, and hence a for-all sparsifier of H with accuracy parameter $\varepsilon' = 1/\log^2 n$ gives the exact value of these cuts in H. We note that the total number of edges that participate in at least one of the 1.1-approximate non-singleton cuts is $O(n)$ [43], which is interesting because there could be as many as $O(n^2)$ such cuts. Hence, we can store all these $O(n)$ edges to get the exact cut values of the approximate minimum cuts in $G \setminus H$. Putting the two things together, we obtain the exact value of all 1.1-approximate minimum cuts and thus obtain the exact minimum cut value in G (along with all the minimum cuts).

Faster Runtime

In the above discussion, our algorithm for $(1+\varepsilon)$ min-cut in worst-case streams has an $\widetilde{O}(n^3)$ post-processing time, and the algorithm for exact min-cut in a random-order stream has an $\widetilde{O}(n^3)$ update time. The difficulty here is related: when enumerating all the $O(1)$-approximate minimum cuts, we need $O(n)$ time to evaluate the cost for each specific cut. We thus propose a general algorithmic framework to overcome this, which is based on the recursive contraction algorithm in [31] (Section 4.4). Namely, when enumerating all the approximate minimum cuts in the recursion tree, we simultaneously maintain a low-space sketch of the columns of the edge-vertex incidence matrix of the corresponding graph. Specifically, we apply a Johnson-Lindenstrauss sketch in our first case and a sparse recovery sketch in our second case, which reduces the evaluation time $O(n)$ to $O(\log n/\varepsilon^2)$ and polylog(n), respectively.

3.2 Open Problems

While we resolve several gaps between the upper and lower bounds, we discuss a remaining open problem. There exist fully dynamic streaming algorithms for approximate minimum cut using $\widetilde{O}(n/\varepsilon^2)$ space where the stream is allowed to both add or delete edges. From Theorem 12, we know that algorithms solving approximate minimum cut in insertion-only streams must use $\Omega(n/\varepsilon)$ space. Therefore, the space complexity of approximate minimum cut in a dynamic stream lies somewhere within these two values, and we believe resolving this gap is a problem of theoretical interest.

▶ **Open Question 15.** What is the exact space complexity of calculating a $(1+\varepsilon)$ approximation to the minimum cut of simple weighted graphs in a one-pass dynamic stream?

We do not have an exact conjecture about this, but we provide an alternative and fully self-contained proof to Theorem 12 in the full version of the paper [17], for which the techniques could help prove a lower bound for the fully dynamic case.

4 Spectral Sparsification and Minimum Cut in Worst-Case Streams

A key result of the section is our construction of Lemma 10, a single-pass insertion-only streaming algorithm for a $(1+\varepsilon)$-for-each cut sparsifier in undirected weighted graphs in $\widetilde{O}(n/\varepsilon)$ space and $n^{o(1)}$ update time. In Section 4.1 and 4.2, we first give our description of graphical spectral sketches (i.e., spectral sparsifiers that are reweighted subgraphs) and online leverage score sampling, which we use as subroutines in our final algorithm. In Section 4.3, we give our full algorithm and prove the correctness of our algorithm. In Section 4.4 and 4.6, we prove the two applications of our algorithm: streaming approximate minimum cut and all-pairs effective resistances.

4.1 Graphical Spectral Sketches

Our algorithm uses the following result in [14] as a subroutine.

▶ **Lemma 16** ([14]). *For $\varepsilon \in (0,1]$, there is an algorithm, which given G, runs in time $m^{1+o(1)}$, and with high probability returns a $(1+\varepsilon)$-for-each spectral sparsifier with $n^{1+o(1)}/\varepsilon$ edges.*

As pointed out by the work of [41], in the above algorithm, the extra $n^{o(1)}$ factor in the number of edges is due to the nearly-linear time short cycle decomposition used in the whole algorithm. We use the simpler and improved short-cycle decomposition algorithm of [41],

which outputs in $m^{1+o(1)}$ time a collection of edge-disjoint cycles of length $O(\log^2 n)$ that cover all but $O(n \log n)$ edges of the graph. Naïvely, the algorithm is also implemented in $m^{1+o(1)}$ space, but we show that a simple modification achieves near-linear space, as follows:

▶ **Lemma 17.** *There is an $m^{1+o(1)}$ time, $\widetilde{O}(m)$ space algorithm that outputs a collection of edge-disjoint cycles of length $O(\log^2 n)$ that cover all but $O(n \log n)$ edges of the graph.*

Proof. The algorithm in Section 3 of [41] first computes a *low-congestion cycle cover*, which is a collection of short cycles that cover most of the edges such that each edge belongs to a small number of cycles. The precise parameters are specified in the statement of Lemma 2 of the full version of [41]: compute a collection of cycles of length $d = O(2^{1/\varepsilon} \log n)$ that cover all but $O(n \log n)$ edges such that each edge appears on at most $c = 1/\varepsilon \cdot n^{O(\varepsilon)}$ cycles. For $\varepsilon = 1/\log\log n$, we obtain $d = O(\log^2 n)$ and $c = n^{o(1)}$. To obtain the short cycle decomposition, the proof of Lemma 2 describes a simple procedure of greedily selecting a subset of edge-disjoint cycles that cover an $\Omega(1/(dc))$ fraction of the edges in the cycle cover and then iterating on the uncovered edges.

Note that the low-congestion cycle cover consists of up to cm edges since each edge can appear in c cycles. Since $c = n^{o(1)}$, storing the entire cycle cover takes up too much space. Our key insight is to terminate the collection of cycles in each iteration of algorithm ImprovedShortCycleDecomp (see Figure 3 in the full version of [41]) once the cycle cover has a total of $O(m)$ edges. Terminating this subroutine early cannot increase the congestion compared to not terminating, thus we maintain the guarantee that the total set of cycles has congestion c. Using the above greedy selection of edge-disjoint cycles, we cover an $\Omega(1/(dc))$ fraction of the edges in our cycle cover, which is $\Omega(1/(dc))$ of all uncovered edges in the graph. Iterating on the uncovered edges gives the same iterations as before up to logarithmic factors. ◀

With our improved short-cycle decomposition, we get the following lemma:

▶ **Lemma 18.** *Given $\varepsilon \in (0, 1]$, there is an algorithm we denote* SpectralSketch*(G, ε), which given G, runs in time $m^{1+o(1)}$ and space $\widetilde{O}(m)$, and with high probability returns a $(1+\varepsilon)$-for-each spectral sparsifier of G with $\widetilde{O}(n/\varepsilon)$ edges.*

4.2 Online Leverage Score Sampling

For the purposes of a faster runtime, we consider only sampling (and reweighting) a fraction of edges during the stream. Existing online leverage score sampling methods allow us to choose edges in an online stream without retracting our choices such that our final graph has $\widetilde{O}(n/\varepsilon^2)$ edges and is a spectral-sparsifier of the original graph. Indeed, let $B_n \in \mathbb{R}^{\binom{n}{2} \times n}$ be the vertex edge incidence matrix of an undirected, unweighted complete graph on n vertices, where the e-th row b_e for edge $e = (u, v)$ has a 1 in column u, a (-1) in column v, and zeroes elsewhere. Then for an arbitrary undirected graph G, we can write its vertex edge incidence matrix $B = SB_n$ where $S \in \mathbb{R}^{\binom{n}{2} \times \binom{n}{2}}$ is a diagonal matrix with entry $\sqrt{w_e}$ in the e-th diagonal entry where w_e is the weight of edge e. It is well-known that the Laplacian matrix L of graph G can be written as $L = B^\top B$. Hence, if we can sample a subset of the rows of B to form a new matrix C (which corresponds to a subset of the edges in G) such that for every x, $\|Bx\|_2 \approx_\varepsilon \|Cx\|_2$, we then have $x^\top Lx \approx_\varepsilon x^\top C^\top Cx$ so that the subgraph H corresponding to C is a $(1+\varepsilon)$-spectral sparsifier of G. For more details, we refer the reader to [27, 15]. Formally, we have the following lemma.

▶ **Lemma 19** (Corollary 2.4 from [15]). *Let G be an undirected simple graph with n vertices and poly(n) bounded edge weights, and $\varepsilon \in (0,1)$. We can construct a $(1+\varepsilon)$-spectral sparsifier of G as a reweighted subgraph with $O(n \log^2 n/\varepsilon^2)$ edges, using only $O(n \log^2 n)$ bits of working memory in the online model. Additionally, the total running time is near-linear in the number of edges of G.*

4.3 Main Algorithm

Our algorithm is presented in Algorithm 1, which is based on the merge-and-reduce paradigm (see, for example, [10]) and uses an additional factor of polylog(n) space. At a high level, our algorithm maintains a number of blocks of edges $\mathbf{B}_0, \mathbf{B}_1, \ldots \mathbf{B}_{\log(n/\varepsilon)}$, each with size $\mathsf{m}_{\mathsf{space}} = \widetilde{O}(n/\varepsilon)$. The most recent edges are stored in $\mathbf{B}_0$; whenever $\mathbf{B}_0$ is full, the successive non-empty blocks $\mathbf{B}_0, \ldots, \mathbf{B}_i$ are merged and reduced to a new graph with $\mathsf{m}_{\mathsf{space}}$ edges, which will be stored in $\mathbf{B}_{i+1}$. Next, we show the correctness of our algorithm. Following a similar argument in [10], we first show the following lemma.

Algorithm 1 StreamingSpectralSparsifier(G, ε).

Input: Undirected graph $G(V,E)$ in a stream with n vertices and m edges, accuracy parameter $\varepsilon \in (\frac{1}{n}, 1)$, offline spectral-sparsifier subroutine SpectralSketch
Output: A graph H with weights w.
Initialize $\mathsf{m}_{\mathsf{space}} = n\log^c(n)/\varepsilon$ for some constant c, $\mathbf{B}_i \leftarrow \emptyset$
foreach *sampled edge e_t (Lemma 19) with rescaled weight u_t* **do**
  if $\mathbf{B}_0$ *does not contain* $\mathsf{m}_{\mathsf{space}}$ *edges* **then**
    $\mathbf{B}_0 \leftarrow e_t \cup \mathbf{B}_0$;
  else
    Let $i > 0$ be the minimal index such that $\mathbf{B}_i = \emptyset$;
    $\mathbf{B}_i, w_i \leftarrow \textsc{SpectralSketch}\left(\mathbf{M}, \frac{\varepsilon}{\log(n/\varepsilon)}\right)$, where $\mathbf{M} = \mathbf{B}_0 \cup \cdots \cup \mathbf{B}_{i-1}$;
    for $j = 0$ **to** $j = i-1$ **do**
      $\mathbf{B}_j \leftarrow \emptyset$;
    $\mathbf{B}_0 \leftarrow e_t$;
$\mathbf{B} \leftarrow \textsc{SpectralSketch}(\mathbf{B}_{\log(n/\varepsilon)} \cup \cdots \cup \mathbf{B}_0, \varepsilon)$
return $\mathbf{B}$.

▶ **Lemma 20** (see Lemma 5.2 in [10]). *Suppose that $\mathbf{B}_0, \ldots, \mathbf{B}_{i-1}$ are all empty while $\mathbf{B}_i$ is non-empty. Then $\mathbf{B}_i$ is a $(1+\frac{\varepsilon}{\log(n/\varepsilon)})^i$-for-each spectral sparsifier for the last $2^{i-1}\mathsf{m}_{\mathsf{space}}$ edges.*

Proof. We prove this by induction on $i \geq 0$. Recall that $\mathbf{B}_i$ can only be non-empty if at some point $\mathbf{B}_0$ contains $\mathsf{m}_{\mathsf{space}}$ and $\mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_{i-1}$ are all non-empty. By induction, suppose that for every $1 \leq j < i$, $\mathbf{B}_j$ is a $(1+\frac{\varepsilon}{\log(n/\varepsilon)})^j$-for-each spectral sparsifier for $2^{j-1}\mathsf{m}_{\mathsf{space}}$ edges. Then we have that $\mathbf{B}_i$ is a $(1+\frac{\varepsilon}{\log(n/\varepsilon)})$-for-each spectral sparsifier for $\mathbf{B}_0 \cup \mathbf{B}_1 \cup \ldots \cup \mathbf{B}_{i-1}$. From the mergeability property of spectral sparsifier graphs, we get that $\mathbf{B}_i$ is a $(1+\frac{\varepsilon}{\log(n/\varepsilon)})(1+\frac{\varepsilon}{\log(n/\varepsilon)})^{i-1} = (1+\frac{\varepsilon}{\log(n/\varepsilon)})^i$-for-each spectral sparsifier for the $\mathsf{m}_{\mathsf{space}} + \sum_{j=0}^{i-2} 2^j \mathsf{m}_{\mathsf{space}} = 2^{i-1}\mathsf{m}_{\mathsf{space}}$ edges, as needed. ◀

Proof of Lemma 10. The success probability of each call to subroutine SpectralSketch is at least $1 - 1/\text{poly}(n)$, hence we can assume that each call to this subroutine and the online leverage score sampling procedure is successful with high probability after taking a union

bound. It then follows from Lemma 20 that $\mathbf{B}_{\log(n/\varepsilon)} \cup \cdots \cup \mathbf{B}_0$ is a $(1 + \frac{\varepsilon}{\log(n/\varepsilon)})^{\log(n/\varepsilon)} \leq (1+\varepsilon)$-for-each spectral sparsifier of the original graph G. Thus $\mathbf{B}$ is a $(1 + O(\varepsilon))(1+\varepsilon) = (1 + O(\varepsilon))$-for-each spectral sparsifier of the original graph G.

Space Complexity. Next, we analyze the space complexity of our algorithm. As stated in Lemma 19, the online leverage score sampling procedure can be implemented in $O(n \log^2 n)$ bits of working memory. We maintain $\log(n/\varepsilon)$ blocks $\mathbf{B}_i$, each taking at most $O(\mathsf{m}_{\mathsf{space}})$ words of space. From Lemma 16 we have that each call to the subroutine SpectralSketch takes at most $\widetilde{O}(\mathsf{m}_{\mathsf{space}} \cdot \log(n/\varepsilon))$ words of space, as the total number of edges never exceeds $\mathsf{m}_{\mathsf{space}} \cdot \log(n/\varepsilon)$. Hence, the total space of the algorithm is $\widetilde{O}(n \log^2 n + \mathsf{m}_{\mathsf{space}} \log(n/\varepsilon)) = \widetilde{O}(n/\varepsilon)$.

Time Complexity. We finally analyze the time complexity of our algorithm. As stated in Lemma 19, the online leverage score sampling procedure can be done in time $\widetilde{O}(m)$, and after the sampling procedure, there are at most $O(n \log^2 n/\varepsilon^2)$ edges. For the merge-and-reduce procedure, each edge participates in at most $\log(n/\varepsilon)$ different spectral sparsifiers, one in block $\mathbf{B}_i$. Additionally, from Lemma 16 we have that SpectralSketch runs in $m^{1+o(1)}$ time where $m = \widetilde{O}(n/\varepsilon)$. We deduce that our total runtime is $\widetilde{O}(m + (n \log^2 n/\varepsilon^2)^{1+o(1)}) = \widetilde{O}(m) + (n/\varepsilon^2)^{1+o(1)}$. ◀

4.4 Approximate Minimum Cut Streaming Algorithm

Our main application of Lemma 10 is the one-pass streaming algorithm for a $(1+\varepsilon)$-approximation to the minimum cut. To achieve this, first recall that we can get a $(1+\varepsilon)$ for-all sparsifier of G in a one-pass stream in nearly-linear time, using $\widetilde{O}(n/\varepsilon^2)$ bits of space (Lemma 4).

The next lemma bounds the number of approximate minimum cuts.

▶ **Lemma 21** ([29]). *For constant $\alpha > 0$, the number of α-approximate minimum cuts is $O(n^{\lfloor 2\alpha \rfloor})$.*

Motivated by [2], we run our algorithm in Lemma 10 along with the streaming for-all sparsifier algorithm ALG in Lemma 4 with accuracy parameter $\varepsilon' = O(1)$. Suppose that the output of ALG is H. We then use H to find the cuts that have a size less than 1.5 times that of the minimum cut value (which can be done in $\text{poly}(n)$ time, see, e.g., [29]). Then, since the number of these approximate minimum cuts is $O(n^3)$, using our $(1+\varepsilon)$ for-each sparsifier, we can get a $(1+\varepsilon)$-approximation to each of these cut values with high probability after taking a union bound. Picking the cut with the minimum value gives us a $(1+\varepsilon)$-approximate minimum cut.

Faster Post-processing Time

The above algorithm is simple but, unfortunately, takes $O(n^3)$ post-processing time. To achieve a faster post-processing time, we consider the recursive contraction algorithm introduced by [31], which enumerates all α-approximate minimum cuts in time $O(n^{2\alpha} \log^2 n)$. We combine this with a Johnson-Lindenstrauss sketch to achieve $\widetilde{O}(n^2/\varepsilon^2)$ post-processing time, proving Theorem 9.

First, we state the lemma from [31]:

▶ **Lemma 22.** *There is an algorithm that can find all the α-approximate minimum cuts with high probability in $O(n^{2\alpha} \log^2 n)$ time and $\widetilde{O}(m)$ space.*

Below, we first give a brief explanation of the algorithm. At a high level, in each step, the algorithm randomly samples an edge proportional to its weights and then contracts the two nodes of the edge. When there are only two remaining nodes, we get a partition of the nodes corresponding to a specific cut of the graph. The work of [31] shows that if we repeat the algorithm $\text{poly}(n)$ times, we can finally enumerate all approximate minimum cuts with high probability. The above process is simple but needs $\text{poly}(n)$ runtime. To make the algorithm faster, [31] then proposes a recursive implementation of the contraction process. Basically, at each level of the recursion, the algorithm reduces the number of nodes by contraction by a constant factor and repeats the algorithm twice in each recursion level if the number of remaining nodes is $\Omega(1)$. Then, each leaf node of the recursion tree corresponds to a specific cut of the graph.

The technical difficulty in our case is evaluating the value of the corresponding cut in a short time when we enter a leaf node in the above process. To achieve this, recall that for a given vertex set $S \subseteq V$, the cut value between S and $V \setminus S$ is $x_S^\top L x_S = x_S^\top B^T B x_S = \|B x_S\|_2^2$ where x_S is the binary indicator vector of S and B is the edge-vertex matrix of the graph (see Section 4.2 for more details). The key observation here is since we only care about a $(1+\varepsilon)$-approximate value, we can apply a JL matrix T to Bx_S, where the matrix T has only $O(\log(n/\varepsilon)/\varepsilon^2)$ rows where for every $x \in \mathbb{R}^{O(n/\varepsilon)}$, $\|Tx\|_2^2 \approx_\varepsilon \|x\|_2^2$ with high probability.

Our algorithm procedure is described as follows. As before, after the stream, we first get a for-all sparsifier K with the accuracy parameter $\varepsilon' = 1/\log n$ (Lemma 4) and a $(1+\varepsilon)$ for-each sparsifier H of the original graph G (Lemma 10). We then use the algorithm in Lemma 22 to enumerate all the approximate minimum cuts of K with $\alpha = 1 + c/\log n$. When maintaining the recursion process of the contraction algorithm, we also maintain a sketch of the columns of the edge-vertex matrix B of H. Specifically, we initially compute a sketch of TB where T is a JL matrix with $O(\log n/\varepsilon^2)$ rows. Then, in the contraction process, when we contract a node pair (u, v), we also merge the u, v-th columns of the sketch with their sums. Note that in this procedure, when we enter a leaf node, there will only remain two columns of our sketch, and the two vectors are exactly TBx_S and $TBx_{V\setminus S}$, for which we can directly compute the cut value in $O(\log n/\varepsilon^2)$ time.

Next, we analyze the time and space complexity in the above procedure. Since the sketch has $O(\log n/\varepsilon^2)$ rows at each recursion level, we need $\widetilde{O}(N \log n/\varepsilon^2)$ words of space to save the sketch, where N is the number of the remaining nodes in the current recursion level. Combining two rows also takes $\widetilde{O}(\log n/\varepsilon^2)$ time. Recall that we need $O(n)$ space in each level and $O(N)$ time to contract two nodes in the original recursive contraction algorithm. This implies after the modification, the space complexity of the above algorithm procedure remains the same of $\widetilde{O}(m) = \widetilde{O}(n/\varepsilon)$, and the time complexity becomes $\widetilde{O}(n^2/\varepsilon^2)$. Putting everything together, we get the correctness of our Theorem 9.

4.5 Approximate Minimum Cut Streaming Lower Bound

To prove this lower bound, we consider the k-edge-connectivity problem as follows. The *edge-connectivity* of a graph is the minimum number of edges that need to be deleted to disconnect the graph. The k-edge-connectivity problem asks whether the edge connectivity of a graph is $< k$ or $\geq k$. The work of [49] shows that any deterministic algorithm that solves the k-edge-connectivity problem in insertion-only streams needs $\Omega(kn \log n)$ bits of space. We note that the proof in [49] also works for randomized algorithms with success probability at least $1 - 1/n$, which implies a lower bound of $\Omega(kn)$ bits of space for any randomized algorithm with constant probability of success.

▷ Claim 23 ([49]). Any randomized algorithm solving the k-edge-connectivity problem in an insertion-only stream with probability at least $2/3$ requires $\Omega(kn)$ bits of space. Moreover, any deterministic algorithm solving the k-edge-connectivity problem in an insertion-only stream requires $\Omega(kn \log n)$ bits.

We now prove the randomized and deterministic lower bounds of Theorem 12.

Randomized Lower Bound

▶ **Lemma 24.** *Fix $\varepsilon > 1/n$. Any randomized algorithm that outputs a $(1+\varepsilon)$-approximation to the minimum cut of a simple, undirected graph in a single pass over a stream with probability at least 2/3 requires $\Omega(n/\varepsilon)$ bits of space*

Proof. Consider the lower bound for k-edge-connectivity in [49] with $k = \frac{1}{10\varepsilon}$ (Claim 23). Distinguishing between whether the graph is $< k$-connected or $\geq k$-connected needs $\Omega(kn) = \Omega(n/\varepsilon)$ space. This implies that deciding whether the min-cut has value $< k$ or $\geq k$ also requires $\Omega(kn) = \Omega(n/\varepsilon)$ space. Suppose that there is a $(1+\varepsilon)$-approximation algorithm for the minimum cut value, in the case where the edge connectivity $\lambda < k$, the approximate value is at most $(1+\varepsilon)\lambda \leq (1+1/10k) \cdot (k-1) < k - 9/10$. In the case where the connectivity $\lambda \geq k$, the approximate value is at least $(1-\varepsilon)\lambda \geq (1-1/10k) \cdot k > k - 1/10$. This implies that an algorithm that gives a $(1+\varepsilon)$-approximation to minimum cut value, solves k-edge-connectivity for $k = \frac{1}{10\varepsilon}$. Thus, we obtain a lower bound of $\Omega(n/\varepsilon)$ bits of space for $(1+\varepsilon)$-approximating the value of the minimum cut. ◀

In addition to the proof above that considers the k-edge-connectivity problem, we also provide an alternative self-contained construction for the randomized result using the Index communication problem in the full version of the paper [17].

Deterministic Lower Bound

If the minimum cut value algorithm is deterministic, then we deterministically solve k-edge-connectivity for $k = \frac{1}{10\varepsilon}$. This gives us a lower bound of $\Omega(kn \log n) = \Omega(n \log n/\varepsilon)$ bits of space for deterministic algorithms. We show how to improve this lower bound to $\Omega(n/\varepsilon^2)$ bits for $\varepsilon \geq n^{-1/4}$. Note that this assumption on ε is consistent with the previous work of [11]. They also use this assumption on ε and prove a lower bound of $\Omega(n \log n/\varepsilon^2)$ bits for cut sparsifiers.

▶ **Lemma 25.** *Fix $\varepsilon > 1/n^{1/4}$. Any deterministic algorithm that outputs a $(1+\varepsilon)$-approximation to the minimum cut value of a simple, undirected graph in a single pass over a stream requires $\Omega(n/\varepsilon^2)$ bits of space*

Proof. A randomized insertion-only streaming algorithm for a for-all cut sparsifier needs $\Omega(n/\varepsilon^2)$ bits of space [2] (since any data structure that stores this information needs that much space). If we carefully look at the proof of [2] in section 3.1, their hard instance is a disjoint union of $\varepsilon^2 n/2$ bipartite graphs with $2/\varepsilon^2$ vertices each. They use the cut sparsifier to query cuts that have vertices only in one of these disjoint graphs, which implies that the cut sparsifier only needs to preserve the cut values for cuts that have at most $1/\varepsilon^4$ edges. Thus, a $(1+\varepsilon)$ cut sparsifier that preserves the cut values of all cuts of size at most $1/\varepsilon^4$ edges needs $\Omega(n/\varepsilon^2)$ bits of space.

We will now show how to simulate a $(1+\varepsilon)$ cut sparsifier that preserves the cut values of all cuts of size at most $1/\varepsilon^4$ using an algorithm for $(1+\varepsilon)$-approximate minimum cut. This will give us the desired lower bound. Let A be a deterministic insertion-only streaming algorithm for $(1+\varepsilon)$-approximate minimum cut value we run during the stream for input graph G.

After the stream, consider any cut S whose value we want to approximate within a $(1+\varepsilon)$ factor (note that the value is at most $1/\varepsilon^4$). We now add some extra edges to the stream for A. We first construct two cliques C_1 and C_2 on $2n$ vertices each and add their edges to the stream. We then add edges between all vertices of S and C_1 and similarly for all vertices in $\bar{S}$ and C_2. Call this new graph G'. We now end the stream and look at the output of A. We claim this is the $(1+\varepsilon)$-approximate value of cut S in the original graph.

To prove this, we must show that S is the minimum cut in the modified graph G'. We know that the value of S is at most $1/\varepsilon^4 \leq n$ (since $\varepsilon \geq n^{-1/4}$). Consider any cut that separates vertices in $C_1 \cup S$. Such a cut has a size of at least $2n$. The same applies to any cut separating vertices in $C_2 \cup \bar{S}$. Thus, the minimum cut is $C_1 \cup S$. None of the new edges we added cross this cut, so the size of this cut remains unchanged in G', implying that we get a $(1+\varepsilon)$-approximation to the cut value in G. Note that the algorithm A has to be deterministic for us to do this because we have to repeat this process for exponentially many cuts (potentially 2^n).

The space taken to build the sparsifier for insertion-only streams is the space taken for A when run on $5n$ vertices. This implies that any deterministic insertion-only streaming algorithm for $(1+\varepsilon)$-approximate minimum cut value needs $\Omega(n/\varepsilon^2)$ bits of space, proving the lower bound for deterministic algorithms. ◀

Proof of Theorem 12. The proof of the theorem directly results from combining Lemma 24 and Lemma 25. Notably, this shows that the spectral sparsifier result in Lemma 10 is optimal in space complexity up to polylogarithmic factors. ◀

4.6 Application: All-Pairs Effective Resistances

Another application of our streaming algorithm from Lemma 10 is calculating all-pairs effective resistances in graph streams. Suppose we have a $(1+\varepsilon)$ for-each spectral sparsifier H of the original graph G. Then, we may hope it will be helpful to approximate the pseudoinverse of G. In fact, in the work of [24], it has been shown that if we have a $(1+\varepsilon)$ for-each spectral sparsifier, we can use it to generate an effective resistance sketch in near-linear time, which can compute the all-pairs effective resistances in $\widetilde{O}(n^2/\varepsilon)$ additional time (Algorithm 8 in [24]). From this, we immediately see the correctness of our Corollary 11.

Lower Bound

We next consider lower bounds. In Theorem 13, we show that if a sketch can approximate each effective resistance with constant probability, then such a sketch must have size at least $\Omega(n/\varepsilon)$. We need the following lemmas about effective resistances and random graphs.

▶ **Lemma 26** (see, e.g., [51]). *For a graph G with unit weight for each edge, we have $C(s,t) = 2mr_{\text{eff}}(s,t)$, where $C(s,t)$ is the commute time of s and t, which is defined as the expected number of steps of a random walk to go from s to t and back.*

▶ **Lemma 27** ([40]). *Let $G(n,p)$ be an Erdos-Renyi random graph. Then we have, with high probability that for any pair $w \neq v$,*

$$h(w,v) = \frac{2|E|}{\deg(v)} + \left\{ \begin{array}{ll} -1 & \text{if } (w,v) \in E \\ -1 + 1/p & \text{if } (w,v) \notin E \end{array} \right\} + O\left(\frac{(\log n)^{3/2}}{\sqrt{n}}\right),$$

where $h(s,t)$ is the hitting time of s,t, defined as the expected number of steps of a random walk from s to t. Note that we have $C(s,t) = h(s,t) + h(t,s)$.

Our reduction is based on the following Index problem.

▶ **Lemma 28** (Index Lower Bound [34]). *Suppose that Alice has a random string $u \in \{0,1\}^n$ and Bob has a random index $i \in [n]$. If Alice sends a single message to Bob from which Bob can recover u_i with probability at least $2/3$, where the probability is over both the randomness of the protocol and the input, then Alice must send $\Omega(n)$ bits to Bob.*

At a high level, we will reduce the Index problem to the effective resistance problem. Suppose Alice has a string $s \in \{0,1\}^{\Theta(n/\varepsilon)}$. We will construct a graph G to encode s such that Bob can recover s_i with constant probability from a $(1 \pm \varepsilon)$ effective resistance sketch of G. By the communication complexity lower bound of the Index problem (Lemma 28), the cut sketch must use $\Omega(n/\varepsilon)$ bits.

Proof of Theorem 13. We first give our construction of the graph G.

Construction of G. We use a bipartite graph G to encode s. Let L and R be the left and right nodes of G where $|L| = |R| = n/2$. We partition L into $O(n\varepsilon)$ disjoint blocks $L_1, \ldots, L_{O(n\varepsilon)}$ of the same size $\frac{1}{\varepsilon}$, and similarly, we partition R into $R_1, \ldots, R_{O(n\varepsilon)}$. We divide s into $n\varepsilon$ disjoint strings $s_i \in \{0,1\}^{(\frac{1}{\varepsilon})^2}$ of the same length. We will encode s_i using the edges from L_i to R_i, where we use G_i to denote the subgraph (L_i, R_i). In particular, if the $s_i(j) = 1$, we form an edge that connects the corresponding node pair in L_i and R_i with unit weight. Note that each G_i is disconnected from the other G_j in this construction.

Recovering a bit in s from an effective resistance sketch of G. Suppose Bob wants to recover a specific bit of s, which belongs to the sub-string $z = s_i$ and has an index t in z. This coordinate corresponds to whether there is an edge between $u \in L_i$ and $v \in R_i$.

Next, we consider the effective resistance between u and v for these two cases. Note that since G_i is disconnected from the other G_j, we just need to consider the sub-graph G_i. By construction, G_i is a random graph with $p = \frac{1}{2}$. From Lemma 26 we get that we only need to show both the hitting time $h(u,v)$ and $h(v,u)$ has a $(1+\varepsilon)$-separation (then the $r_{\text{eff}}(u,v)$) will also have a $(1+\varepsilon)$-separation. Then, from Lemma 27, we get that both the $h(u,v)$ and $h(v,u)$ will have a $\Theta(1)$ gap for the two cases, and with high probability, we have that both of $h(u,v)$ and $h(v,u)$ are $\Theta(1/\varepsilon)$. From this, we can get that $r_{\text{eff}}(u,v)$ will have a $(1+\varepsilon)$-separation for the two cases (note that Alice can also send the degree of each node and the number of edges in each G_i to Bob, which only needs $O(n\log(1/\varepsilon))$ bits), which yields an $\Omega(n/\varepsilon)$ lower bound of the effective resistance sketch size. ◀

5 Exact Minimum Cut in Random-Order Streams

In this section, we give an $\widetilde{O}(n)$ algorithm that outputs the minimum cut in a simple, unweighted graph in a single-pass random-order stream. Recall that in a simple graph, there is at most one edge between any pair of vertices. For a clearer demonstration, we will present an initial construction for an algorithm with an $\widetilde{O}(n^2)$ update time when an edge arrives. Then, we shall show how to improve the update time to $\widetilde{O}(n)$, proving Theorem 14.

5.1 Initial Construction

We will use the algorithm in Lemma 4 (for-all sparsifier in a stream) as a subroutine. The whole algorithm is given in Algorithm 2. Below, we first give a high-level explanation of our algorithm. We first consider the case when the minimum cut size is $s = \Omega(\log n)$. Since we are now considering the random-order model, a prefix of the edges gives us partial information about the cut sizes. Let H be the subgraph of G which is formed by a prefix of the edges of the stream where $|H| \approx \frac{|G| \log n}{s}$ and $|G|$, $|H|$ are the number of the edges in G and H respectively. By a Chernoff bound, one can show that with high probability, for every subset $S \subset V$, $w_H(S, V \setminus S)$ is a small constant approximation to $w_G(S, V \setminus S)$. H might be too large to store in memory, so we store H_1, a $(1+\varepsilon)$-for-all sparsifier of H, during the stream (we will decide ε later). Therefore, using the graph H_1, we know all cuts in G whose cut size is within a factor of 1.1 of the true minimum cut of G. After this step, when a new edge $e \in G \setminus H$ arrives, we can use graph H_1 to check whether e belongs to some 1.1-approximate minimum cut in G, and if so, we save this edge (from Lemma 30 we know that the total number of these edges saved is at most $O(n)$). The next question is how to estimate the exact value of these 1.1-approximate cuts in H. The crucial observation is since the minimum cut size of H is an integer and is at most $\Theta(\log n)$, we can set the approximation parameter $\varepsilon = 1/\log^2 n$ for H_1. This gives the exact value of $w_H(S, V \setminus S)$ if S and $V \setminus S$ is a constant-approximate minimum cut in G (since the error is at most $O(\log n) \cdot 1/\log^2 n < 1$). Putting these things together, after the stream, we can enumerate all 1.1-approximate minimum cuts and obtain their exact values, thus getting the exact minimum cut value.

The remaining case to handle is when the minimum cut size is $O(\log n)$. Similarly, in this case, setting $\varepsilon = 1/\log^2 n$ will give us the exact value of the minimum cut. Finally, we do not know the value of s, so we will try all powers of 2 for it. Note that we do not need the exact value; a 2-approximation to s will suffice. The detailed description of the algorithm is given in Algorithm 2.

To show the correctness of our algorithm, we first prove the following key lemma.

▶ **Lemma 29.** *Suppose that H is a subgraph of G formed by a prefix of the edges in the random order model. We have that for every $S \subset V$, if $|H| = |G| \cdot \frac{\ell}{w_G(S, V \setminus S)}$, then with probability at least $1 - 2e^{-c_2 \cdot \ell}$, $|w_H(S, V \setminus S) - \ell| \leq 0.1\ell$.*

Proof. Consider each edge e that is a cut edge between S and $V \setminus S$ (let T denote the set of such edges). Let y_e denote the indicator random variable where $y_e = 1$ if $e \in H$ and $y_e = 0$ otherwise. Then we have that $\mathbb{E}[w_H(S, V \setminus S)] = \mathbb{E}[\sum_{e \in T} y_e] = \ell$ from the assumption in the random-order model. By a standard Chernoff bound, we have that

$$\Pr[|w_H(S, V \setminus S) - \ell| \geq 0.1\ell] \leq 2e^{-c_2 \cdot \ell}$$

for some constant c_2. ◀

We also need the following structural result for the approximate minimum cuts of a graph.

▶ **Lemma 30** ([43])**.** *For any simple, unweighted graph and any constant $\varepsilon > 0$, the total number of edges that participate in non-singleton $(2-\varepsilon)$-approximate minimum cuts is at most $O(n)$.*

We first consider the case when the minimum cut size $s < c \log n$. In this case, the minimum cut of H will never exceed $c \log n$ since H is a subgraph of G. Since the cut value is an integer, setting $\varepsilon = 1/\log^2 n$ gives the exact value of the size of the minimum cut.

Algorithm 2 MinCutRandomOrder.

Input : Undirected and unweighted graph $G(V, E)$ in a random-order stream with n vertices and m edges.

ALG_1 is an instance of Lemma 4 with $\varepsilon = 1/\log^2 n$.
Maintain the degree of each node d_i during the stream.
foreach *edge e in the graph stream* **do**
    Feed e to ALG_1 ;
    if *e is the 2^i-th edge for some $i \in \mathbb{N}$* **then**
        $H_1 \leftarrow$ output of ALG_1 (Here H_1 is a for-all sparsifier of the prefix graph H);
        if *the minimum cut size of H_1 is larger than $c \log n$* **then**
            Save H_1 and break the for loop.
$H_1 \leftarrow$ output of ALG_1 ;
if *there is no new edge in the stream* **then**
    return *the minimum cut value of H_1.*
$T \leftarrow \emptyset$.
foreach *new edge e during the stream* **do**
    if *e is the cut edge between some S and $V \setminus S$ where S and $V \setminus S$ is a non-singleton 1.1-approximate minimum cut in H_1* **then**
        Add e to T
foreach *S where S and $V \setminus S$ is a non-singleton 1.1-approximate minimum cut in H_1* **do**
    $v_S \leftarrow w_{H_1}(S, V \setminus S) + |\{e \in T : e \text{ is a cut edge between } S \text{ and } V \setminus S\}|$
return *the minimum value of* $\min v_S$ *and* $\min d_i$

We next consider the other case when $s \geq c \log n$. We will first show that, as the number of edges in H increases, the minimum cut size of H will increase to $\Theta(\log n)$ when $|H| = \Theta\big(\frac{|G| \log n}{s}\big)$. We next analyze the value of $w_H(S, V \setminus S)$. Let $\mathcal{S}_i$ $(1 \leq i \leq \log n)$ denote the set of nodes such that

$$\mathcal{S}_i = \{S \subset V \mid s \cdot 2^{i-1} \leq w_G(S, V \setminus S) \leq s \cdot 2^i\}$$

For each $\mathcal{S}_i$, from Lemma 21 we know that $|\mathcal{S}_i| \leq O(n^{2^{i+1}})$ and for every $S \in \mathcal{S}_i$, from Lemma 29 we know that with probability at least $1 - 2e^{-c_2 c \cdot \log n \cdot 2^{i-1}} = 1 - 2n^{-c_2 c \cdot 2^{i-1}}$, $w_H(S, V \setminus S) \cdot \frac{|G|}{|H|}$ is a 1.1-approximation of $w_G(S, V \setminus S)$. Note that the constant c here can be sufficiently large, and after taking a union bound over all $S \in \mathcal{S}_i$ and all $\log n$ $\mathcal{S}_i$ we get that with probability at least 9/10, for every $S \subset V$, $w_H(S, V \setminus S) \cdot \frac{|G|}{|H|}$ is a 1.1-approximation of $w_G(S, V \setminus S)$. From the above analysis we immediately know that when $\frac{|G|}{|H|} = \Theta\big(\frac{s}{\log n}\big)$ from the subgraph H, we can learn the set

$$\mathcal{S} = \{S \subset V \mid s \leq w_G(S, V \setminus S) \leq 1.1^2 s\},$$

and the next step is choosing the true minimum cut among them. For every non-single node set $S \in \mathcal{S}$, as mentioned, since $w_H(S, V \setminus S)$ is $\Theta(\log n)$, setting $\varepsilon = 1/\log^2 n$ we can get its exact value. The remaining step is to estimate the value of $w_{G \setminus H}(S, V \setminus S)$. As described in Algorithm 2, after the previous step, we save every edge $e \in G \setminus H$, where e belongs to at least one of the non-singleton 1.1^2-approximate minimum cuts in $\mathcal{S}$. We know the exact

value of $w_{G\setminus H}(S, V \setminus S)$ from this edge set. Putting the two things together, for every $S \in \mathcal{S}$ and $|S| \geq 2$, we know the value of $w_G(S, V \setminus S)$ after taking a sum of the two parts. For every singleton cut, we simply maintain the degree of each node during the stream. Thus, after taking the minimum value of the two parts (singleton and non-singleton cuts), we get the exact value of the minimum cut.

Space Complexity

In the first phase of the algorithm, we use one instance of the algorithm in Lemma 4 with $\varepsilon = 1/\log^2 n$, which takes space $\widetilde{O}(n)$. In the second phase of the algorithm, we save all the edges that belong to the approximate minimum cut in H. From Lemma 30, we know that there are at most $O(n)$ such edges, and hence this part takes $O(n)$ words of space. We also maintain the degree of each node during the stream, which takes $O(n)$ words of space. Putting everything together, the space usage of our algorithm is $\widetilde{O}(n)$.

Time Complexity

In the first phase of the algorithm, when one edge e comes, the update and query time of ALG_1 is polynomial, and we can also find the minimum cut in polynomial time. Hence, the update time here is still polynomial. In the second phase, when one edge e comes, we can enumerate all 1.1-approximate minimum cuts to check whether e belongs to one of the approximate minimum cuts (we can use $O(n^2)$ time to enumerate all approximate minimum cuts, see, e.g., [29]). Hence, we can do the update of this step in polynomial time. After all edges come, we can similarly enumerate all minimum cuts, from which the overall algorithm can be implemented in polynomial time.

5.2 Faster Update and Post-Processing Time

In the above algorithm, the runtime bottleneck is when a non-prefix edge e comes; we need to enumerate all approximate minimum cuts in the prefix graph to check whether to keep this edge. To get a faster runtime, we instead do a check when we collect n edges and consider a similar procedure to what we did in Section 4.4 where we use the recursive contraction algorithm with parameter $\alpha = 1 + \frac{1}{\log n}$. The difference is when doing the recursive contraction algorithm, we maintain the sketch SB where S is a k-sparse recovery matrix (see, e.g., [20]) with $k \log(n/k)$ rows where $k = O(\log n)$ and B is the edge-vertex matrix of the n edges we currently collected. Particularly, during the recursive contraction process, when we contract the nodes u, v, we merge the columns that u, v corresponds to in the sketch SB and replace them with their sum. Next, we consider each leaf node in the recursion, corresponding to one specific cut for the prefix graph. We can check whether it is a non-singleton and approximate minimum cut in $O(1)$ time from the information the algorithm keeps. Note that if without the sketch matrix S, for each edge e we want to check, it belongs to this specific cut if and only if the e-th coordinates of the remaining two columns of the edge-vertex matrix are 1 and -1, and otherwise these two coordinates are both 0. Since the minimum cut of the prefix graph is $\Theta(\log n)$, we can get that both of the remaining columns are $O(\log n)$-sparse before multiplying the sketch matrix S. This means that a k-sparse approximate recovery algorithm with $k = O(\log n)$ is sufficient to recover the indices of the non-zero coordinates of the remaining columns, which helps us to find the corresponding edges.

Time and Space Complexity

We analyze the time and space complexity of the above procedure. At each level of the recursion, since the sketch has $k\log(n/k)$ rows, we need $\widetilde{O}(Nk)$ words of space to save the sketch, and it takes $\widetilde{O}(k)$ time to combine two rows, where N is the number of the remaining nodes in the current level of the recursion. Recall that we need $O(n)$ space in each level and $O(N)$ time to contract two nodes in the original recursive contraction algorithm. This implies the modification only increases the time and space complexity by a factor of $\widetilde{O}(k)$. Since the decoding time of the k-sparse recovery sketch is $k \cdot \text{polylog}(n)$ and $k = O(\log n)$, we have that the procedure has time complexity $\widetilde{O}(n^2)$ and space complexity $\widetilde{O}(n)$. Finally, note that when we collect a set of n edges in the suffix graph, we can do our above checking procedure while collecting the following n edges in the graph stream, which results in a strictly $O(n)$ update time of our algorithm. Similarly, suppose the minimum cut of the graph is $\Theta(c)$. When we do the post-processing, we can use the c-sparse recovery sketch when enumerating all the approximate minimum cuts, which results in a $\widetilde{O}(n^2c)$ post-processing time. Putting everything together, we have the correctness of Theorem 14.

Lastly, we note that our algorithm can not only return the exact min-cut value but also collect all the edges that participate in any of the minimum cuts. When computing the for-all sparsifier of the prefix graph, for an edge that participates in at least one approximate min-cut, the probability that it will be sampled is $\Omega(1)$ as otherwise, the estimated cut value in such a cut will have two different values each with a constant probability, contradicting the guarantees of the for-all sparsifier. Hence, we can collect all of these edges from oversampling by a constant factor, which results in finding all the minimum cuts of the input graph and the edges crossing them.

References

1 Noga Alon. On the edge-expansion of graphs. *Combinatorics, Probability and Computing*, 6(2):145–152, 1997. URL: `http://journals.cambridge.org/action/displayAbstract?aid=46517`.

2 Alexandr Andoni, Jiecao Chen, Robert Krauthgamer, Bo Qin, David P. Woodruff, and Qin Zhang. On sketching quadratic forms. In Madhu Sudan, editor, *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 311–319, 2016. `doi:10.1145/2840728.2840753`.

3 Sepehr Assadi, MohammadHossein Bateni, Aaron Bernstein, Vahab Mirrokni, and Cliff Stein. Coresets meet edcs: algorithms for matching and vertex cover on massive graphs. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1616–1635. SIAM, 2019. `doi:10.1137/1.9781611975482.98`.

4 Sepehr Assadi and Soheil Behnezhad. Beating Two-Thirds For Random-Order Streaming Matching. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:13, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ICALP.2021.19`.

5 Sepehr Assadi and Aditi Dudeja. *A Simple Semi-Streaming Algorithm for Global Minimum Cuts*, pages 172–180. Society for Industrial and Applied Mathematics, January 2021. `doi:10.1137/1.9781611976496.19`.

6 Joshua D. Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-Ramanujan sparsifiers. *SIAM J. Comput.*, 41(6):1704–1721, 2012. `doi:10.1137/090772873`.

7 András A. Benczúr and David R. Karger. Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 47–55. ACM, 1996. `doi:10.1145/237814.237827`.

8 András A. Benczúr and David R. Karger. Randomized approximation schemes for cuts and flows in capacitated graphs. *SIAM J. Comput.*, 44(2):290–319, 2015. doi:10.1137/070705970.

9 Aaron Bernstein. Improved bounds for matching in random-order streams. *Theory of Computing Systems*, pages 1–15, 2023.

10 Vladimir Braverman, Petros Drineas, Cameron Musco, Christopher Musco, Jalaj Upadhyay, David P. Woodruff, and Samson Zhou. Near optimal linear algebra in the online and sliding window models. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 517–528. IEEE, 2020. doi:10.1109/FOCS46700.2020.00055.

11 Charles Carlson, Alexandra Kolla, Nikhil Srivastava, and Luca Trevisan. Optimal lower bounds for sketching graph cuts. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2565–2569. SIAM, 2019. doi:10.1137/1.9781611975482.158.

12 Amit Chakrabarti, Graham Cormode, and Andrew McGregor. Robust lower bounds for communication and stream computation. *Theory of Computing*, 12(10):1–35, 2016. doi:10.4086/toc.2016.v012a010.

13 Amit Chakrabarti, Prantar Ghosh, Andrew McGregor, and Sofya Vorotnikova. Vertex ordering problems in directed graph streams. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1786–1802. SIAM, 2020. doi:10.1137/1.9781611975994.109.

14 Timothy Chu, Yu Gao, Richard Peng, Sushant Sachdeva, Saurabh Sawlani, and Junxing Wang. Graph sparsification, spectral sketches, and faster resistance computation, via short cycle decompositions. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 361–372. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00042.

15 Michael B. Cohen, Cameron Musco, and Jakub Pachocki. Online row sampling. *Theory of Computing*, 16(15):1–25, 2020. doi:10.4086/toc.2020.v016a015.

16 Artur Czumaj, Hendrik Fichtenberger, Pan Peng, and Christian Sohler. Testable properties in general graphs and random order streaming. *arXiv preprint*, 2019. arXiv:1905.01644.

17 Matthew Ding, Alexandro Garces, Jason Li, Honghao Lin, Jelani Nelson, Vihan Shah, and David P. Woodruff. Space complexity of minimum cut problems in single-pass streams, 2024. arXiv:2412.01143.

18 Alireza Farhadi, Mohammad Taghi Hajiaghayi, Tung Mah, Anup Rao, and Ryan A Rossi. Approximate maximum matching in random streams. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1773–1785. SIAM, 2020.

19 Buddhima Gamlath, Sagar Kale, Slobodan Mitrovic, and Ola Svensson. Weighted matchings via unweighted augmentations. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pages 491–500, 2019. doi:10.1145/3293611.3331603.

20 Anna C. Gilbert, Yi Li, Ely Porat, and Martin J. Strauss. Approximate sparse recovery: Optimizing time and measurements. *SIAM J. Comput.*, 41(2):436–453, 2012. doi:10.1137/100816705.

21 Sudipto Guha and Andrew McGregor. Stream order and order statistics: Quantile estimation in random-order streams. *SIAM Journal on Computing*, 38(5):2044–2059, 2009. doi:10.1137/07069328X.

22 Diba Hashemi and Weronika Wrzos-Kaminska. Weighted matching in the random-order streaming and robust communication models. *arXiv preprint*, 2024. doi:10.48550/arXiv.2408.15434.

23 Julien M. Hendrickx, Karl Henrik Johansson, Raphaël M. Jungers, Henrik Sandberg, and Kin Cheong Sou. Efficient computations of a security index for false data attacks in power networks. *IEEE Transactions on Automatic Control*, 59(12):3194–3208, 2014. doi:10.1109/TAC.2014.2351625.

24 Arun Jambulapati and Aaron Sidford. Efficient õ (n/eps) spectral sketches for the laplacian and its pseudoinverse. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2487–2503. SIAM, 2018.

25 Michael Kapralov. Space lower bounds for approximating maximum matching in the edge arrival model. In *Proceedings of the Thirty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '21, pages 1874–1893, USA, 2021. Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611976465.112.

26 Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Approximating matching size from random streams. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 734–751. SIAM, 2014. doi:10.1137/1.9781611973402.55.

27 Michael Kapralov, Yin Tat Lee, Cameron Musco, Christopher Musco, and Aaron Sidford. Single pass spectral sparsification in dynamic streams. *SIAM J. Comput.*, 46(1):456–477, 2017. doi:10.1137/141002281.

28 Michael Kapralov, Aida Mousavifar, Cameron Musco, Christopher Musco, Navid Nouri, Aaron Sidford, and Jakab Tardos. Fast and space efficient spectral sparsification in dynamic streams. In *Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '20, pages 1814–1833, USA, 2020. Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611975994.111.

29 David R. Karger. Minimum cuts in near-linear time. *J. ACM*, 47(1):46–76, 2000. doi:10.1145/331605.331608.

30 David R. Karger and Matthew S. Levine. Random sampling in residual graphs. In John H. Reif, editor, *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 63–66. ACM, 2002. doi:10.1145/509907.509918.

31 David R. Karger and Clifford Stein. A new approach to the minimum cut problem. *J. ACM*, 43(4):601–640, 1996. doi:10.1145/234533.234534.

32 Christian Konrad. A simple augmentation method for matchings with applications to streaming algorithms. In *43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018*, pages 74–1. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, 2018.

33 Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 231–242. Springer, 2012. doi:10.1007/978-3-642-32512-0_20.

34 Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1996. doi:10.1017/CBO9780511574948.

35 Aleksander Madry. Fast approximation algorithms for cut-based problems in undirected graphs. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 245–254. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.30.

36 Andrew McGregor. Graph stream algorithms: A survey. *SIGMOD Rec.*, 43(1):9–20, May 2014. doi:10.1145/2627692.2627694.

37 Morteza Monemizadeh, Shan Muthukrishnan, Pan Peng, and Christian Sohler. Testable bounded degree graph properties are random order streamable. *arXiv preprint*, 2017. arXiv:1707.07334.

38 Sagnik Mukhopadhyay and Danupon Nanongkai. Weighted min-cut: sequential, cut-query, and streaming algorithms. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 496–509. ACM, 2020. doi:10.1145/3357713.3384334.

39 J.I. Munro and M.S. Paterson. Selection and sorting with limited storage. *Theoretical Computer Science*, 12(3):315–323, 1980. doi:10.1016/0304-3975(80)90061-4.

40 Andrea Ottolini and Stefan Steinerberger. Concentration of hitting times in erdős-rényi graphs. *Journal of Graph Theory*, 2023.

41 Merav Parter and Eylon Yogev. Optimal short cycle decomposition in almost linear time. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019*, Leibniz International Proceedings in Informatics, LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, July 2019. doi:10.4230/LIPIcs.ICALP.2019.89.

42 Pan Peng and Christian Sohler. Estimating graph parameters from random order streams. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2449–2466. SIAM, 2018. doi:10.1137/1.9781611975031.157.

43 Aviad Rubinstein, Tselil Schramm, and S. Matthew Weinberg. Computing exact minimum cuts without knowing the graph. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 39:1–39:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICS.ITCS.2018.39.

44 Jonah Sherman. Breaking the multicommodity flow barrier for o(vlog n)-approximations to sparsest cut. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 363–372. IEEE Computer Society, 2009. doi:10.1109/FOCS.2009.66.

45 Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM J. Comput.*, 40(6):1913–1926, 2011. doi:10.1137/080734029.

46 Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 81–90. ACM, 2004. doi:10.1145/1007352.1007372.

47 Daniel A. Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM J. Comput.*, 40(4):981–1025, 2011. doi:10.1137/08074489X.

48 Satoshi Sugiura and Fumitaka Kurauchi. Isolation vulnerability analysis in road network: Edge connectivity and critical link sets. *Transportation Research Part D: Transport and Environment*, 119:103768, 2023.

49 Xiaoming Sun and David P. Woodruff. Tight Bounds for Graph Problems in Insertion Streams. In Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX-/RANDOM 2015)*, volume 40 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 435–448, Dagstuhl, Germany, 2015. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.APPROX-RANDOM.2015.435.

50 Xinjue Wang, Ke Deng, Jianxin Li, Jeffery Xu Yu, Christian S Jensen, and Xiaochun Yang. Efficient targeted influence minimization in big social networks. *World Wide Web*, 23(4):2323–2340, 2020. doi:10.1007/S11280-019-00748-Z.

51 David P. Williamson. Lecture notes in spectral graph theory, October 2016. URL: https://people.orie.cornell.edu/dpw/orie6334/Fall2016/lecture13.pdf.

52 Mariano Zelke. Intractability of min- and max-cut in streaming graphs. *Information Processing Letters*, 111(3):145–150, 2011. doi:10.1016/J.IPL.2010.10.017.

Learning-Augmented Streaming Algorithms for Approximating MAX-CUT

Yinhao Dong
School of Computer Science and Technology, University of Science and Technology of China, Hefei, China

Pan Peng
School of Computer Science and Technology, University of Science and Technology of China, Hefei, China

Ali Vakilian
Toyota Technological Institute at Chicago (TTIC), IL, USA

Abstract

We study learning-augmented streaming algorithms for estimating the value of MAX-CUT in a graph. In the classical streaming model, while a 1/2-approximation for estimating the value of MAX-CUT can be trivially achieved with $O(1)$ words of space, Kapralov and Krachun [STOC'19] showed that this is essentially the best possible: for any $\epsilon > 0$, any (randomized) single-pass streaming algorithm that achieves an approximation ratio of at least $1/2 + \epsilon$ requires $\Omega(n/2^{\mathrm{poly}(1/\epsilon)})$ space.

We show that it is possible to surpass the 1/2-approximation barrier using just $O(1)$ words of space by leveraging a (machine learned) oracle. Specifically, we consider streaming algorithms that are equipped with an ϵ-accurate oracle that for each vertex in the graph, returns its correct label in $\{-1, +1\}$, corresponding to an optimal MAX-CUT solution in the graph, with some probability $1/2 + \epsilon$, and the incorrect label otherwise.

Within this framework, we present a single-pass algorithm that approximates the value of MAX-CUT to within a factor of $1/2 + \Omega(\epsilon^2)$ with probability at least 2/3 for insertion-only streams, using only $\mathrm{poly}(1/\epsilon)$ words of space. We also extend our algorithm to fully dynamic streams while maintaining a space complexity of $\mathrm{poly}(1/\epsilon, \log n)$ words.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Learning-Augmented Algorithms, Graph Streaming Algorithms, MAX-CUT

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.44

Related Version *Full Version*: https://arxiv.org/abs/2412.09773

Funding The work was conducted in part while Pan Peng and Ali Vakilian were long-term visitors at the Simons Institute for the Theory of Computing as part of the Sublinear Algorithms program. Yinhao Dong and Pan Peng are supported in part by NSFC grant 62272431.

1 Introduction

Given an undirected, unweighted graph, the MAX-CUT problem seeks to find a partition of the vertices (a cut) that maximizes the number of edges with endpoints on opposite sides of the partition (such edges are said to be "cut" by the partition). MAX-CUT is a well-known NP-hard problem. The best-known approximation algorithm, developed by Goemans and Williamson [22], achieves a 0.878-approximation ratio, which is the best possible under the Unique Games Conjecture [33].

We consider the problem of estimating the value of MAX-CUT in a graph under the streaming model of computation, where MAX-CUT denotes some fixed optimal solution, and the value of a solution is defined as the number of edges in the graph that are cut by the partition. In this model, the graph is presented as a sequence of edge insertions and


© Yinhao Dong, Pan Peng, and Ali Vakilian;
licensed under Creative Commons License CC-BY 4.0

deletions, known as a graph stream. The objective is to analyze the graph's structure using minimal space. The model is called insertion-only if the stream contains only edge insertions; if both insertions and deletions are allowed, it is referred to as the dynamic model.

In the streaming model, a 1/2-approximation for estimating the value of MAX-CUT can be trivially achieved using $O(1)$ words of space, where a word represents the space required to encode the size of the graph. This is done by simply counting the total number m of edges in the graph and outputting $m/2$. However, Kapralov and Krachun [31] showed that this is essentially the best possible: for any $\epsilon > 0$, any (randomized) single-pass streaming algorithm in the insertion-only model that achieves an approximation ratio of at least $1/2 + \epsilon$ requires $\Omega(n/2^{\text{poly}(1/\epsilon)})$ space.

We focus on the problem of estimating the value of MAX-CUT in a streaming setting within the framework of *learning-augmented algorithms*. These algorithms utilize predictions from a machine learning model to solve a problem, where the predictions typically include some information about the optimal solution, future events, or yet unread data in the input stream (see, e.g., [24, 27, 13]). Learning-augmented algorithms have gained significant attention recently, partly due to numerous breakthroughs in machine learning. Ideally, such algorithms should be both *robust* and *consistent*: when the predictions are accurate, the algorithm should outperform the best-known classical algorithms, and when the predictions are inaccurate, the algorithm should still provide performance guarantees that are close to or match those of the best-known classical algorithms. Despite the extensive research in the area of learning-augmented algorithms over the past few years[1], our understanding of this framework within the streaming model remains limited.

In this work, we consider the *noisy prediction* model, also referred to as *ϵ-accurate predictions*, where the algorithm has oracle access to a prediction vector $Y \in \{-1, 1\}^n$. Each entry Y_v is *independently correct* with probability $\frac{1}{2} + \epsilon$, where $\epsilon \in (0, \frac{1}{2}]$ represents the *bias* of the predictions. Specifically, for each $v \in V$, we have $\Pr[Y_v = x^*_v] = \frac{1}{2} + \epsilon$ and $\Pr[Y_v = -x^*_v] = \frac{1}{2} - \epsilon$, where $x^* \in \{-1, 1\}^n$ denotes some fixed but unknown optimal solution for MAX-CUT. This model captures a scenario where an ML algorithm (or oracle) provides predictions for the values of x^* that are unreliable and noisy, being only slightly better than random guesses (i.e., just above the 1/2 probability that a random solution would satisfy). Recently, variants of this prediction model have been used to design improved approximation algorithms for MAX-CUT and constraint satisfaction problems (CSPs) [14, 21].

Specifically, we study the following question:

> *Given an oracle $\mathcal{O}$ that provides ϵ-accurate predictions Y about the optimal* MAX-CUT *of a streaming graph G, can we improve upon the worst-case approximation ratio and space trade-off established by [31], specifically the $(\frac{1}{2} + \epsilon)$ ratio with $\Omega(n)$ space complexity, for estimating the* MAX-CUT *value?*

1.1 Our Results

We provide an affirmative answer by presenting a single-pass streaming algorithm that surpasses the 1/2-approximation barrier for the value of MAX-CUT using only $\text{poly}(1/\epsilon)$ (resp., $\text{poly}(1/\epsilon, \log n)$)words of space in insertion-only (resp., fully dynamic) streams. Formally, we establish the following result:

[1] See `https://algorithms-with-predictions.github.io/` for an up-to-date repository of publications on this topic.

▶ **Theorem 1.** *Let $\epsilon \in (0, \frac{1}{2}]$. Given oracle access to an ϵ-accurate predictor, there exists a single-pass streaming algorithm that provides a $\left(\frac{1}{2} + \Omega(\epsilon^2)\right)$-approximation for estimating the MAX-CUT value of a graph in insertion-only (resp., fully dynamic) streams using* $\mathrm{poly}(1/\epsilon)$ *(resp.,* $\mathrm{poly}(1/\epsilon, \log n)$*) words of space with probability at least* $2/3$*.*

By using median trick, we can boost the success probability to $1 - \delta$, where $\delta \in (0, 1)$, with an additional $\log(1/\delta)$ factor in space complexity.

We remark that the "robustness" of our learning-augmented algorithm comes for free, as we can always compute the number of edges in the graph and report half of them as a $\frac{1}{2}$-approximation solution, regardless of the value of ϵ. Furthermore, with predictions, it gains an advantage of $\Omega(\epsilon^2)$ in the approximation ratio. Furthermore, our algorithm does not require the exact value of ϵ; a constant-factor approximation of ϵ is sufficient. This estimation is only needed to determine the sample size.

Our algorithm is based on the observation that when the maximum degree of a graph is relatively small (i.e. smaller than $\mathrm{poly}(\epsilon) \cdot m$), the number of edges with endpoints having different predicted labels already provides a strictly better-than-$\frac{1}{2}$ approximation of the MAX-CUT value. For general graphs, we employ some well-known techniques, such as the CountMin sketch and ℓ_0-sampling, to distinguish between high-degree and low-degree vertices in both insertion-only and dynamic streams. By combining a natural greedy algorithm with an algorithm tailored for the low-degree part, we achieve a non-trivial approximation. The space complexity of the resulting algorithm is primarily determined by the subroutines for identifying the set of high-degree vertices, which can be bounded by $\mathrm{poly}(1/\epsilon, \log n)$.

We note that it is standard to assume that the noisy oracle $\mathcal{O}$ has a compact representation and that its space complexity is not included in the space usage of our algorithm, following the conventions in streaming learning-augmented algorithms [24, 27, 13, 1]. Indeed, as noted in [24], a reliable oracle can often be learned space-efficiently in practice as well. Moreover, we show that in the case of random-order streams, our algorithm only needs to query $\mathcal{O}$ for the labels of a constant number of vertices (see Section B).

Additionally, our algorithm actually works in a weaker model. That is, we can assume that in our streaming framework, the predicted label of each vertex is associated with its edges. Thus, the elements in a stream can be represented as $(e = (u, v), Y_u, Y_v)$, where the predictions $(Y_u)_{u \in V}$ remain consistent throughout the stream. The case of dynamic streams is defined analogously. Notably, in this model, no additional space is required to store the predictors.

1.2 Related Work

Learning-augmented algorithms have been actively researched in online algorithms [42, 35, 4, 6, 7, 25, 41, 5, 36], data structures [43, 20, 46, 39, 44], graph algorithms [18, 12, 8, 37, 16, 10, 40, 23, 17, 14, 21, 11], and sublinear-time algorithms [26, 19, 38, 45]. Our algorithms fit into the category of learning-augmented streaming algorithms. Hsu et al. [24] introduced learning-augmented streaming algorithms for frequency estimation, and Jiang et al. [27] extended this framework to various norm estimation problems in data stream. Recently, Aamand et al. [2, 1] developed learning-augmented frequency estimation algorithms, that improve upon the work of [24]. Additionally, Chen et al. [13] studied single-pass streaming algorithms to estimate the number of triangles and four-cycles in a graph. It is worth mentioning that both our work and previous efforts on learning-augmented streaming algorithms mainly focus on using predictors to improve the corresponding space-accuracy trade-offs. Furthermore, recent studies have explored learning-augmented algorithms for MAX-CUT and constraint

satisfaction problems (CSPs) [14, 21] within a variant of our prediction model. In particular, for the Max-CUT problem, Cohen-Addad et al. [14] achieved a $(0.878+\tilde{\Omega}(\epsilon^4))$-approximation using SDP. However, it is important to emphasize that their setting differs significantly from ours: they focus on finding the MAX-CUT in the offline setting, whereas our goal is to estimate its size in the streaming setting. Additionally, Ghoshal et al. [21] developed an algorithm that yields near-optimal solutions when the average degree is larger than a threshold determined only by the bias of the predictions, independent of the instance size. They further extended this result to weighted graphs and the MAX-3LIN problem.

The aforementioned lower bound proven in [31] is, in fact, the culmination of a series of works exploring the trade-offs between space and approximation ratio for the streaming MAX-CUT value problem, including [29, 34, 30]. In our main context, we focus on the problem of estimating the value of MAX-CUT in a streaming setting. Another related problem is that of finding the MAX-CUT itself in a streaming model. Since even outputting the MAX-CUT requires $\Omega(n)$ space, research in this area primarily considers streaming algorithms that use $\tilde{O}(n)$ space, known as the semi-streaming model, where $\tilde{O}(\cdot)$ hides polylogarithmic factors. Beyond the trivial $\frac{1}{2}$-approximation algorithm, which simply partitions the vertices randomly into two parts, there exists a $(1-\epsilon)$-approximation algorithm for finding MAX-CUT. This algorithm works by constructing a $(1-\epsilon)$-cut or spectral sparsifier of the input streaming graph, which can be achieved in $\tilde{O}(n)$ space (see e.g. [3, 32]), and then finding the MAX-CUT of the sparsified graph (which may require exponential time).

2 Preliminaries and Problem Statement

Notations. Throughout the paper we let $G=(V,E)$ be an undirected, unweighted graph with n vertices and m edges. Given a vertex $v \in V$ and disjoint sets $S,T \subseteq V$, $e(v,S)$ denotes the number of edges incident to v whose other endpoint belongs to S; $e(v,S) = |\{(u,v) \mid (u,v) \in E \text{ and } u \in S\}|$. Similarly, we define $e(S,T)$ to denote the number of edges with one endpoint in S and the other endpoint in T; $e(S,T) = \sum_{v \in T} e(v,S)$. A cut (S,T) of G is a bipartition of V into two disjoint subsets S and T. The value of the cut (S,T) is $e(S,T)$.

In the descriptions of our algorithms, for a vertex set $S \subseteq V$, we define S^+ (resp. S^-) as the set of vertices with predicted $+$ (resp. $-$) signs by the given ϵ-accurate oracle $\mathcal{O}$. Note that S^+ and S^- form a bipartition of S. Throughout the paper, all space complexities are measured in words unless otherwise specified. The space complexity in bits is larger by a factor of $\log m + \log n$.

In our algorithms, we utilize several well-known techniques in the streaming model, which we define below.

CountMin Sketch [15]. Given a stream of m integers from $[n]$, let f_i denote number of occurrences of i in the stream, for any $i \in [n]$. In CountMin sketch, there are k uniform and independent random hash functions $h_1, h_2, \ldots, h_k : [n] \to [w]$ and an array C of size $k \times w$. The algorithm maintains C, such that $C[\ell, s] = \sum_{j:h_\ell(j)=s} f_j$ at the end of the stream. Upon each query of an arbitrary $i \in [n]$, the algorithm returns $\tilde{f}_i = \min_{\ell \in [k]} C[\ell, h_\ell(i)]$.

▶ **Theorem 2** (CountMin Sketch [15]). *For any $i \in [n]$ and any $\ell \in [k]$, we have $\tilde{f}_i \geq f_i$ and $\mathbb{E}[C[\ell, h_\ell(i)]] \leq f_i + \frac{m}{w}$. The space complexity is $O(kw)$ words. Let $\epsilon, \delta \in (0,1)$. If we set $k = \lceil \frac{e}{\epsilon} \rceil$ and $w = \lceil \ln \frac{1}{\delta} \rceil$, then for any $i \in [n]$, we have $\tilde{f}_i \geq f_i$, and with probability at least $1-\delta$, $\tilde{f}_i \leq f_i + \epsilon m$. The corresponding space complexity is $O(\frac{1}{\epsilon} \ln \frac{1}{\delta})$ words.*

Reservoir Sampling [47]. The algorithm samples k elements from a stream, such that at any time $m \geq k$, the sample consists of a uniformly random subset of size k of the elements seen so far. The space complexity of the algorithm is $O(k(\log n + \log m))$ bits.

▶ **Definition 3** (ℓ_0-sampler [28]). *Let $x \in \mathbb{R}^n$ be a non-zero vector and $\delta \in (0,1)$. An ℓ_0-sampler for x returns FAIL with probability at most δ and otherwise returns some index i such that $x_i \neq 0$ with probability $\frac{1}{|\operatorname{supp}(x)|}$ where $\operatorname{supp}(x) = \{i \mid x_i \neq 0\}$ is the support of x.*

The following theorem states that ℓ_0-samplers can be maintained using a single pass in dynamic streams.

▶ **Theorem 4** (Theorem 2 in [28]). *Let $\delta \in (0,1)$. There exists a single-pass streaming algorithm for maintaining an ℓ_0-sampler for a non-zero vector $x \in \mathbb{R}^n$ (with failure pribability δ) in the dynamic model using $O(\log^2 n \log \delta^{-1})$ bits of space.*

Problem Statement. In this work, we consider the problem of estimating the value of MAX-CUT of a graph G in the learning-augmented streaming model. Formally, given an undirected, unweighted graph $G = (V, E)$, which is represented as a sequence of edges, i.e., $\sigma := \langle e_1, e_2, \dots \rangle$ for $e_i \in E$, our goal is to scan the sequence in one pass and output an estimate of the MAX-CUT value, while minimizing space usage. When the sequence contains only edge insertions, it is referred to as an *insertion-only* stream. When the sequence contains both edge insertions and deletions, it is referred to as a *dynamic* stream. Note that in dynamic streams, the sequence of the stream is often represented as $\sigma := \langle (e_1, \Delta_1), (e_2, \Delta_2), \dots \rangle$, where for each i, $e_i \in E$ and $\Delta_i = 1$ (resp. -1) denotes edge insertion (resp. deletion).

Furthermore, we assume that the algorithms are equipped with an oracle $\mathcal{O}$, which provides ϵ-accurate predictions $Y \in \{-1, 1\}^n$ where $\epsilon \in (0, \frac{1}{2}]$. This information is provided via an external oracle, and for the purpose of our algorithm, we assume that the edges in the stream are annotated with the predictions of their endpoints.

Specifically, for each vertex $v \in V$, $\Pr[Y_v = x^*_v] = \frac{1}{2} + \epsilon$ and $\Pr[Y_v = -x^*_v] = \frac{1}{2} - \epsilon$, where $x^* \in \{-1, 1\}^n$ is some fixed but unknown optimal solution. Following previous works on MAX-CUT with ϵ-accurate predictions [14, 21], we also assume that the $(Y_v)_{v \in V}$ are independent.

Finally, since we assume that accuracy parameter of predictions, ϵ, is known to the algorithm up to a constant factor in advance, and the space complexity of all algorithms in the paper is at most $\operatorname{poly}(\log n, 1/\epsilon, 1/\delta)$, we will assume throughout that $m = \Omega(\epsilon^{-11}\delta^{-7})$. Otherwise, when $m = O(\epsilon^{-11}\delta^{-7})$, we can simply store all edges in $\operatorname{poly}(1/\epsilon, 1/\delta)$ space and compute the MAX-CUT exactly.

3 The Simple Case of Low-Degree Graphs

In this section, we show that when all vertices have "low-degree" then following the ϵ-accurate predictions, we can beat the $\frac{1}{2}$-approximation barrier for streaming MAX-CUT. Informally, once an edge (u, v) arrives in the stream, we consult the noisy oracle $\mathcal{O}$ and if their endpoints are predicted to be in different parts by the noisy oracle, i.e., $\mathcal{O}(u) \neq \mathcal{O}(v)$, we increase the size of the cut by one. This can be simply implemented using $O(\log n)$ bits of space. Refer to Algorithm 1 for a formal description.

Next, we analyze the space complexity and approximation guarantee of Algorithm 1.

Algorithm 1 A learning-augmented streaming algorithm for estimating MAX-CUT value in low-degree graphs.

Input: Graph G with $\Delta < \frac{\epsilon^2\delta m}{4} - \frac{1}{4\epsilon^2}$ as a stream of edges, an ϵ-accurate oracle $\mathcal{O} \to \{-1, 1\}$
Output: Estimate of the MAX-CUT value of G

```
initialize X ← 0.
for each edge (u, v) in the stream do
    Y_u = O(u), Y_v = O(v)
    if Y_u ≠ Y_v then X ← X + 1
return X
```

▶ **Theorem 5.** *Let $\epsilon \in (0, \frac{1}{2}]$ and $\delta \in (0, 1)$. Given an ϵ-accurate oracle $\mathcal{O}$ for the MAX-CUT of G, if the maximum degree Δ in G satisfies $\Delta < \frac{\epsilon^2\delta m}{4} - \frac{1}{4\epsilon^2}$, then with probability at least $1 - \delta$, Algorithm 1 outputs a $(\frac{1}{2} + \epsilon^2)$-approximation of the MAX-CUT value of G. The algorithm uses $O(1)$ words of space.*

Proof. We first analyze the space complexity. In Algorithm 1, we maintain a counter X for the edges whose endpoints have different signs, which takes $O(\log n)$ bits of space. Therefore, the space complexity of the algorithm is $O(1)$ words.

Next, we show that Algorithm 1 with high probability outputs a $(\frac{1}{2} + \epsilon^2)$-approximation of the MAX-CUT when all vertices have low degrees.

Since (V^+, V^-) is a feasible cut of G, where V^+ (resp. V^-) is the set of vertices with predicted $+$ (resp. $-$) signs by the given ϵ-accurate oracle $\mathcal{O}$, we have $X \le \mathrm{OPT}$, where OPT denotes the (optimal) MAX-CUT value of G. For each edge $(u, v) \in E$, define an indicator random variable X_{uv} such that $X_{uv} = 1$ if $Y_u \neq Y_v$, and is zero otherwise.

Let x^* be the assignment vector corresponding to the MAX-CUT in G. Specifically, if $(S, V \setminus S)$ is the MAX-CUT of G, then $x^*_v = 1$ if $v \in S$, and -1 otherwise. Consider the following two cases:

(I.1) If $x^*_u \neq x^*_v$, then $\Pr[Y_u \neq Y_v] = \Pr[Y_u = x^*_u \wedge Y_v = x^*_v] + \Pr[Y_u \neq x^*_u \wedge Y_v \neq x^*_v] = \frac{1}{2} + 2\epsilon^2$.
(I.2) If $x^*_u = x^*_v$, then $\Pr[Y_u \neq Y_v] = \Pr[Y_u = x^*_u \wedge Y_v \neq x^*_v] + \Pr[Y_u \neq x^*_u \wedge Y_v = x^*_v] = \frac{1}{2} - 2\epsilon^2$.

Note that $X = \sum_{(u,v)\in E} X_{uv}$ and $\mathbb{E}[X_{uv}] = \Pr[Y_u \neq Y_v]$. Then, by the linearity of expectation,

$$\begin{aligned}\mathbb{E}[X] = \sum_{\substack{(u,v)\in E\\ x^*_u \neq x^*_v}} \mathbb{E}[X_{uv}] + \sum_{\substack{(u,v)\in E\\ x^*_u = x^*_v}} \mathbb{E}[X_{uv}] &= \mathrm{OPT} \cdot (\frac{1}{2} + 2\epsilon^2) + (m - \mathrm{OPT}) \cdot (\frac{1}{2} - 2\epsilon^2)\\ &= (\frac{1}{2} - 2\epsilon^2)m + 4\epsilon^2 \cdot \mathrm{OPT}\\ &\ge (\frac{1}{2} + 2\epsilon^2) \cdot \mathrm{OPT},\end{aligned}$$

where the last inequality holds since $\frac{1}{2} - 2\epsilon^2 \ge 0$ for $\epsilon \in (0, \frac{1}{2}]$ and $m \ge \mathrm{OPT}$.

Next, we bound $\mathrm{Var}[X]$. Note that $\mathrm{Var}[X_{uv}] = \mathbb{E}[X_{uv}] - \mathbb{E}[X_{uv}]^2$ since $(X_{uv})_{(u,v)\in E}$ are indicator random variables. By (I.1) and (I.2), $\mathrm{Var}[X_{uv}] = \frac{1}{4} - 4\epsilon^4$.

(II.1) If $x^*_u \neq x^*_v$, then $\mathrm{Var}[X_{uv}] = \left(\frac{1}{2} + 2\epsilon^2\right) - \left(\frac{1}{2} + 2\epsilon^2\right)^2 = \frac{1}{4} - 4\epsilon^4$.
(II.2) If $x^*_u = x^*_v$, then $\mathrm{Var}[X_{uv}] = \left(\frac{1}{2} - 2\epsilon^2\right) - \left(\frac{1}{2} - 2\epsilon^2\right)^2 = \frac{1}{4} - 4\epsilon^4$.

For any pair of distinct edges $(u,v),(u,w) \in E$, we compute $\mathbb{E}[X_{uv}X_{uw}] = \Pr[X_{uv} = X_{uw} = 1] = \Pr[Y_v = Y_w = -Y_u]$ and $\mathrm{Cov}[X_{uv}, X_{uw}] = \mathbb{E}[X_{uv}X_{uw}] - \mathbb{E}[X_{uv}] \cdot \mathbb{E}[X_{uw}]$.

(III.1) If $x^*_u = x^*_v$ and $x^*_u = x^*_w$, then

$$\begin{aligned}\mathbb{E}[X_{uv}X_{uw}] &= \Pr[Y_u = x^*_u \wedge Y_v \neq x^*_v \wedge Y_w \neq x^*_w] + \Pr[Y_u \neq x^*_u \wedge Y_v = x^*_v \wedge Y_w = x^*_w] \\ &= \left(\frac{1}{2}+\epsilon\right)\left(\frac{1}{2}-\epsilon\right)^2 + \left(\frac{1}{2}-\epsilon\right)\left(\frac{1}{2}+\epsilon\right)^2 = \frac{1}{4}-\epsilon^2.\end{aligned}$$

Therefore, $\mathrm{Cov}[X_{uv}, X_{uw}] = \left(\frac{1}{4}-\epsilon^2\right) - \left(\frac{1}{2}-2\epsilon^2\right)^2 = \epsilon^2 - 4\epsilon^4$.

(III.2) If $x^*_u = x^*_v$ and $x^*_u \neq x^*_w$, then

$$\begin{aligned}\mathbb{E}[X_{uv}X_{uw}] &= \Pr[Y_u = x^*_u \wedge Y_v \neq x^*_v \wedge Y_w = x^*_w] + \Pr[Y_u \neq x^*_u \wedge Y_v = x^*_v \wedge Y_w \neq x^*_w] \\ &= \left(\frac{1}{2}+\epsilon\right)^2\left(\frac{1}{2}-\epsilon\right) + \left(\frac{1}{2}-\epsilon\right)^2\left(\frac{1}{2}+\epsilon\right) = \frac{1}{4}-\epsilon^2.\end{aligned}$$

Therefore, $\mathrm{Cov}[X_{uv}, X_{uw}] = \left(\frac{1}{4}-\epsilon^2\right) - \left(\frac{1}{2}-2\epsilon^2\right)\left(\frac{1}{2}+2\epsilon^2\right) = 4\epsilon^4 - \epsilon^2$.

(III.3) If $x^*_u \neq x^*_v$ and $x^*_u = x^*_w$, then

$$\begin{aligned}\mathbb{E}[X_{uv}X_{uw}] &= \Pr[Y_u = x^*_u \wedge Y_v = x^*_v \wedge Y_w \neq x^*_w] + \Pr[Y_u \neq x^*_u \wedge Y_v \neq x^*_v \wedge Y_w = x^*_w] \\ &= \left(\frac{1}{2}+\epsilon\right)^2\left(\frac{1}{2}-\epsilon\right) + \left(\frac{1}{2}-\epsilon\right)^2\left(\frac{1}{2}+\epsilon\right) = \frac{1}{4}-\epsilon^2.\end{aligned}$$

Therefore, $\mathrm{Cov}[X_{uv}, X_{uw}] = \left(\frac{1}{4}-\epsilon^2\right) - \left(\frac{1}{2}+2\epsilon^2\right)\left(\frac{1}{2}-2\epsilon^2\right) = 4\epsilon^4 - \epsilon^2$

(III.4) If $x^*_u \neq x^*_v$ and $x^*_u \neq x^*_w$, then

$$\begin{aligned}\mathbb{E}[X_{uv}X_{uw}] &= \Pr[Y_u = x^*_u \wedge Y_v = x^*_v \wedge Y_w = x^*_w] + \Pr[Y_u \neq x^*_u \wedge Y_v \neq x^*_v \wedge Y_w \neq x^*_w] \\ &= \left(\frac{1}{2}+\epsilon\right)^3 + \left(\frac{1}{2}-\epsilon\right)^3 = \frac{1}{4}+3\epsilon^2.\end{aligned}$$

Therefore, $\mathrm{Cov}[X_{uv}, X_{uw}] = \left(\frac{1}{4}+3\epsilon^2\right) - \left(\frac{1}{2}+2\epsilon^2\right)^2 = \epsilon^2 - 4\epsilon^4$.

Hence, $\mathrm{Cov}[X_{uv}, X_{uw}] \leq \epsilon^2 - 4\epsilon^4$. So,

$$\begin{aligned}\mathrm{Var}[X] &= \sum_{(u,v)\in E} \mathrm{Var}[X_{uv}] + \sum_{\substack{(u,v),(u,w)\in E \\ v\neq w}} \mathrm{Cov}[X_{uv}, X_{uw}] + \sum_{\substack{(u,v),(w,z)\in E \\ u,v,w,z \text{ all distinct}}} \mathrm{Cov}[X_{uv}, X_{wz}] \\ &\leq m \cdot \left(\frac{1}{4} - 4\epsilon^4\right) + m \cdot \Delta \cdot \left(\epsilon^2 - 4\epsilon^4\right) + 0 \leq \left(\frac{1}{4} + \Delta\epsilon^2\right) m.\end{aligned}$$

Then, by applying Chebyshev's inequality,

$$\begin{aligned}\Pr\left[X \leq \left(\frac{1}{2}+\epsilon^2\right)\mathrm{OPT}\right] &\leq \Pr\left[|X - \mathbb{E}[X]| \geq \epsilon^2 \cdot \mathrm{OPT}\right] \\ &\leq \frac{\mathrm{Var}[X]}{\epsilon^4 \cdot \mathrm{OPT}^2} \\ &\leq \frac{\left(\frac{1}{4}+\Delta\epsilon^2\right)m}{\epsilon^4 \cdot \mathrm{OPT}^2} && \triangleright\ \mathrm{Var}[X] \leq (\frac{1}{4}+\Delta\epsilon^2)m \\ &\leq \frac{4\cdot(\frac{1}{4}+\Delta\epsilon^2)}{\epsilon^4 m} && \triangleright\ \mathrm{OPT} \geq \frac{m}{2} \\ &< \delta. && \triangleright\ \Delta < \frac{\epsilon^2\delta m}{4} - \frac{1}{4\epsilon^2}\end{aligned}$$

So, with probability at least $1-\delta$, $X \geq (\frac{1}{2}+\epsilon^2)\cdot\mathrm{OPT}$. ◀

4 Our Algorithm for General Graphs

In this section, we present our $(\frac{1}{2} + \Omega(\epsilon^2))$-approximation for streaming MAX-CUT in $O(\text{poly}(1/\epsilon, 1/\delta))$ space using ϵ-accurate predictions. For better presenting our algorithmic ideas, we first consider the problem in the *random order* streams in Section 4.1. Next, in Section 4.2, we show how to extend our algorithm to arbitrary order streams using more advanced sketching techniques. Finally, in Section 4.3, we show that our algorithm also work in dynamic streams where both edge insertions and deletions are allowed.

Offline Implementation. We first describe our algorithm in the offline setting. Recall that our algorithm from Section 3 is effective when the maximum degree in the graph is smaller than a pre-specified threshold $\phi = \Theta(\epsilon^2\delta m)$, where δ is the target failing probability of the algorithm. Let H and L respectively denote the high-degree (i.e., vertices with degree at least ϕ) and low-degree (i.e., vertices with degree less than ϕ) vertices in the input graph $G = (V, E)$. Now, by the guarantee of Theorem 5, suppose we have a $(\frac{1}{2} + \epsilon^2)$-approximation of MAX-CUT on the induced subgraph $G[L]$ denoted by (L^+, L^-). This cut is exactly the cut suggested by the given ϵ-accurate oracle $\mathcal{O}$. Next, we extend this cut using the vertices in H in a greedy manner. We pick vertices in H one by one in an arbitrary order and each time add them to either L^+ or L^- sections so that the size of cut maximized. We denote the resulting cut by $(C, V \setminus C)$. Another cut we consider is simply (H, L). We show at least one these two cuts is a $(\frac{1}{2} + \Omega(\epsilon^2))$-approximation for MAX-CUT on G with high probability.

Throughout the paper, we define high-degree vertices as those with degree $\geq \frac{\epsilon^2 m}{c}$, where $c = \frac{80}{\delta}$. Vertices with degrees below this threshold are considered low-degree.

▶ **Theorem 6.** *Let $\epsilon \in (0, \frac{1}{2}]$ and $\delta \in (0, 1)$. The best of $(C, V \setminus C)$ and (H, L) cuts is a $(\frac{1}{2} + \frac{\epsilon^2}{16})$-approximation for MAX-CUT on G with probability at least $1 - \delta$.*

Without loss of generality, we assume that $e(H, L) < (\frac{1}{2} + \epsilon^2) \cdot \text{OPT}$, otherwise we are done. Then it suffices to show that $e(C, V \setminus C) \geq (\frac{1}{2} + \frac{\epsilon^2}{16}) \cdot \text{OPT}$. Recall that the cut $(C, V \setminus C)$ is obtained by extending the cut (L^+, L^-) of $G[L]$ suggested by the given ϵ-accurate oracle using the vertices in H in a greedy manner. We first show that Algorithm 1 works for the induced subgraph $G[L]$.

▶ **Lemma 7.** *Let $\epsilon \in (0, \frac{1}{2}]$ and $\delta \in (0, 1)$. Suppose that $e(H, L) < (\frac{1}{2} + \epsilon^2) \cdot \text{OPT}$, $c = \frac{80}{\delta}$ and $m > \frac{c^3\delta}{\epsilon^4} + \frac{1}{4\epsilon^4}$. Then there exists a $(\frac{1}{2} + \epsilon^2)$-approximation of the MAX-CUT value of $G[L]$ with probability at least $1 - \delta$.*

Proof. Note that $m = m_L + m_H + e(H, L)$, where m_L (resp. m_H) is the number of edges in $G[L]$ (resp. $G[H]$). Then we have $m_L = m - m_H - e(H, L) > m - \frac{4c^2}{\epsilon^4} - (\frac{1}{2} + \epsilon^2)m$, since $|H| \leq \frac{2m}{\epsilon^2 m/c} = \frac{2c}{\epsilon^2}$ and $e(H, L) < (\frac{1}{2} + \epsilon^2) \cdot \text{OPT} \leq (\frac{1}{2} + \epsilon^2)m$.

Since the high-degree threshold is $\frac{\epsilon^2 m}{c}$, we have $\Delta_L < \frac{\epsilon^2 m}{c}$, where Δ_L is the maximum degree of $G[L]$. It is easy to check that $\frac{\epsilon^2 m}{c} < \frac{\epsilon^2\delta}{4}(m - \frac{4c^2}{\epsilon^4} - (\frac{1}{2} + \epsilon^2)m) - \frac{1}{4\epsilon^2}$ by substituting in the conditions for c and m. It follows that $\Delta_L < \frac{\epsilon^2\delta m_L}{4} - \frac{1}{4\epsilon^2}$.

Therefore, by the guarantee of Theorem 5, with probability at least $1 - \delta$, there exists a $(\frac{1}{2} + \epsilon^2)$-approximation of MAX-CUT on $G[L]$ denoted by (L^+, L^-), i.e., $e(L^+, L^-) \geq (\frac{1}{2} + \epsilon^2) \cdot \text{OPT}_L$, where OPT_L is the size of the MAX-CUT value of $G[L]$. ◀

▶ **Lemma 8.** *Let $\epsilon \in (0, \frac{1}{2}]$. Suppose that $e(H, L) = \alpha \cdot \text{OPT}$ (where $\alpha < \frac{1}{2} + \epsilon^2$) and $m > \frac{8c^2}{\epsilon^8}$. We have $\text{OPT}_L > (1 - \alpha - \epsilon^4) \cdot \text{OPT}$, where OPT_L is the size of the MAX-CUT value of $G[L]$.*

Proof. Note that $\mathrm{OPT} \leq \mathrm{OPT}_L + \mathrm{OPT}_H + e(H, L)$, where OPT_L (resp. OPT_H) is the size of the MAX-CUT of $G[L]$ (resp. $G[H]$). It follows that

$$\begin{aligned}
\mathrm{OPT}_L &\geq \mathrm{OPT} - \mathrm{OPT}_H - e(H, L) \\
&\geq \mathrm{OPT} - \frac{4c^2}{\epsilon^4} - e(H, L) && \triangleright\ \mathrm{OPT}_H \leq m_H \leq |H|^2 \leq \frac{4c^2}{\epsilon^4} \\
&= \mathrm{OPT} - \frac{4c^2}{\epsilon^4} - \alpha \cdot \mathrm{OPT} && \triangleright\ e(H, L) = \alpha \cdot \mathrm{OPT} \\
&> \mathrm{OPT} - \epsilon^4 \cdot \mathrm{OPT} - \alpha \cdot \mathrm{OPT} && \triangleright\ \frac{4c^2}{\epsilon^4} < \epsilon^4 \cdot \frac{m}{2} \leq \epsilon^4 \cdot \mathrm{OPT} \\
&= (1 - \alpha - \epsilon^4) \cdot \mathrm{OPT}.
\end{aligned}$$

◀

Proof of Theorem 6. Since the algorithm returns the best of two cuts, if the value of the cut (H, L) is at least $(\frac{1}{2} + \epsilon^2) \cdot \mathrm{OPT}$ (i.e., $e(H, L) \geq (\frac{1}{2} + \epsilon^2) \cdot \mathrm{OPT}$), then we are done. Hence, without loss of generality, we can assume that the size of the cut (H, L) is strictly less than $(\frac{1}{2} + \epsilon^2) \cdot \mathrm{OPT}$. Based on Lemma 7 and Lemma 8, we have

$$\begin{aligned}
e(C, V \setminus C) &\geq e(L^+, L^-) + \sum_{v \in H} \max\{e(v, L^+), e(v, L^-)\} \\
&\geq \left(\frac{1}{2} + \epsilon^2\right) \mathrm{OPT}_L + \sum_{v \in H} \max\{e(v, L^+), e(v, L^-)\} && \triangleright\ e(L^+, L^-) \geq (\tfrac{1}{2} + \epsilon^2)\mathrm{OPT}_L \\
&\geq \left(\frac{1}{2} + \epsilon^2\right) \mathrm{OPT}_L + \frac{1}{2} \cdot e(H, L) && \triangleright\ \max\{e(v, L^+), e(v, L^-)\} \geq \frac{e(v, L)}{2} \\
&> \left(\frac{1}{2} + \epsilon^2\right) (1 - \alpha - \epsilon^4)\mathrm{OPT} + \frac{\alpha}{2} \cdot \mathrm{OPT} && \triangleright\ \mathrm{OPT}_L > (1 - \alpha - \epsilon^4)\mathrm{OPT} \\
&= \left(\frac{1}{2} + (1 - \alpha)\epsilon^2 - \frac{\epsilon^4}{2} - \epsilon^6\right) \mathrm{OPT} \\
&> \left(\frac{1}{2} + \left(\frac{1}{2} - \epsilon^2\right)\epsilon^2 - \frac{\epsilon^4}{2} - \epsilon^6\right) \mathrm{OPT} && \triangleright\ \alpha < \frac{1}{2} + \epsilon^2 \\
&\geq \left(\frac{1}{2} + \frac{\epsilon^2}{16}\right) \mathrm{OPT}, && \triangleright\ \left(\frac{1}{2} - \epsilon^2\right)\epsilon^2 - \frac{\epsilon^4}{2} - \epsilon^6 \geq \frac{\epsilon^2}{16}
\end{aligned}$$

with probability at least $1 - \delta$. ◀

4.1 Warm-up: Random Order Streams

Overview of the Algorithm. Suppose that the edges of $G = (V, E)$ arrive one by one in a random order stream. At a high level, we would like to run our algorithm from Section 3 for low-degree vertices L (i.e., the induced subgraph on L), and then, using a greedy approach, add the high-degree vertices H to the constructed cut. However, the set H and L are not known a priori in the stream, and it is not clear how to store the required information to run the described algorithm, space efficiently.

To detect high-degree vertices and collect sufficient information to run the greedy approach on them at the end of the stream, we rely on the random order of the stream. We store a small number of edges from the beginning of the stream and then gather degree information for those vertices that are candidates for being high-degree. More precisely, we store the first $\mathrm{poly}(1/\epsilon, 1/\delta)$ edges in the (random order) stream and use them to identify a set $\tilde{H}$ of size $\mathrm{poly}(1/\epsilon, 1/\delta)$ that contains all high-degree vertices H with probability $1 - \delta$. We then store all edges between vertices in $\tilde{H}$ throughout the stream, which requires $\mathrm{poly}(1/\epsilon, 1/\delta)$ words of space. Additionally, for every vertex $v \in \tilde{H}$, we maintain the number of edges between v

and $\tilde{L}^+ \coloneqq V^+ \setminus \tilde{H}$, as well as between v and $\tilde{L}^- \coloneqq V^- \setminus \tilde{H}$. It is straightforward to check that these counters require $\text{poly}(1/\epsilon, 1/\delta)$ words of space in total. We then apply Algorithm 1 to the graph $G[V \setminus \tilde{H}]$ to approximate the MAX-CUT value of $G[V \setminus \tilde{H}]$.

Finally, at the end of stream, since we can compute the degree of all vertices in $\tilde{H}$ exactly, we can determine the set of high degree vertices $H \subseteq \tilde{H}$. First, for the remaining vertices $\tilde{S} \coloneqq \tilde{H} \setminus H$, we (hypothetically) feed them to our algorithm for the low-degree vertices. Note that we have all edges between any pair of vertices in $\tilde{S}$, as well as all the number of their incident edges to $\tilde{L}^+$ and $\tilde{L}^-$. Therefore, we can compute the size of (L^+, L^-) cut exactly. Now it only remains to run the greedy algorithm for the high-degree vertices H. Similarly, since we have computed the number of incident edges of high-degree vertices to $\tilde{L}^+, \tilde{L}^-$, and we have stored all edges with both endpoints in $\tilde{H}$, we can perform the greedy extension of $(\tilde{L}^+ \cup \tilde{S}^+, \tilde{L}^- \cup \tilde{S}^-)$ using H. Moreover, using the same set of degree information, we can compute the size of $(H, V \setminus H)$ as well. Then, we can return the best of these two cuts as our estimate of MAX-CUT in G.

Next, we prove the approximation guarantee and the space complexity of the algorithm. The algorithm is formally given in Algorithm 2.

▶ **Theorem 9.** *Let $\epsilon \in (0, \frac{1}{2}]$ and $\delta \in (0, 1)$. Given an ϵ-accurate oracle $\mathcal{O}$ for the MAX-CUT of G, there exists a single-pass $(\frac{1}{2} + \frac{\epsilon^2}{16})$-approximation algorithm for estimating the MAX-CUT value of G in the insertion-only random-order streams. The algorithm uses $O(\frac{1}{\delta^6\epsilon^8})$ words of space. The approximation holds with probability at least $1 - \delta$.*

▶ **Lemma 10.** *Let $\delta \in (0, 1)$. Then $\tilde{H}$ contains all high-degree vertices in G with probability at least $1 - \frac{\delta}{2}$.*

Proof. Suppose that v is a high-degree vertex in G, i.e., $\deg(v) \geq \frac{\epsilon^2 m}{c}$. For each edge e_i incident to v, define an indicator random variable

$$X_i = \begin{cases} 1, & \text{if } e_i \text{ is in the first } \frac{\beta}{\delta^3\epsilon^4} \text{ edges of the random order stream} \\ 0, & \text{otherwise} \end{cases}$$

So, $\mathbb{E}[X_i] = \Pr[X_i = 1] = \frac{\beta}{\delta^3\epsilon^4 m}$. We define $X \coloneqq \sum_{i \in [\deg(v)]} X_i$ to denote the number of edges incident to v that appear in the first $\frac{\beta}{\delta^3\epsilon^4}$ edges of the random order stream. Then, $\mathbb{E}[X] = \sum_{i \in [\deg(v)]} \mathbb{E}[X_i] = \frac{\beta \deg(v)}{\delta^3\epsilon^4 m}$.

Next, we bound $\text{Var}[X]$. For any $i \in [\deg(v)]$, $\text{Var}[X_i] = \mathbb{E}[X_i^2] - (\mathbb{E}[X_i])^2 = \mathbb{E}[X_i] - (\mathbb{E}[X_i])^2 = \frac{\beta}{\delta^3\epsilon^4 m} - (\frac{\beta}{\delta^3\epsilon^4 m})^2 = \frac{\beta}{\delta^3\epsilon^4 m} - \frac{\beta^2}{\delta^6\epsilon^8 m^2}$. Since $(X_i)_{i \in [\deg(v)]}$ are negatively correlated, $\text{Var}[X] \leq \sum_{i \in [\deg(v)]} \text{Var}[X_i] = \deg(v) \cdot \left(\frac{\beta}{\delta^3\epsilon^4 m} - \frac{\beta^2}{\delta^6\epsilon^8 m^2}\right) \leq \frac{\beta \deg(v)}{\delta^3\epsilon^4 m}$. Therefore, by Chebyshev's inequality,

$$\begin{aligned} \Pr[v \notin \tilde{H}] = \Pr[X \leq 0] &= \Pr\left[X \leq \mathbb{E}[X] - \frac{\beta \deg(v)}{\delta^3\epsilon^4 m}\right] \\ &\leq \frac{\text{Var}[X] \cdot \delta^6\epsilon^8 m^2}{\beta^2 \deg^2(v)} \\ &\leq \frac{\delta^3\epsilon^4 m}{\beta \cdot \deg(v)} && \triangleright\ \text{Var}[X] \leq \frac{\beta \deg(v)}{\delta^3\epsilon^4 m} \\ &\leq \frac{\delta^3\epsilon^2 c}{\beta}. && \triangleright\ \deg(v) \geq \frac{\epsilon^2 m}{c} \end{aligned}$$

By our definition of high-degree vertices, there are at most $\frac{2m}{\epsilon^2 m/c} = \frac{2c}{\epsilon^2}$ high-degree vertices in G. By union bound, with probability at least $1 - \frac{2c}{\epsilon^2} \cdot \frac{\delta^3\epsilon^2 c}{\beta} = 1 - \frac{2c^2\delta^3}{\beta} \geq 1 - \frac{\delta}{2}$ (since $c = \frac{80}{\delta}$ and β is sufficiently large), $\tilde{H}$ contains all high-degree vertices in G. ◀

Algorithm 2 Estimating the MAX-CUT value in (insertion-only) random-order streams.

Input: Graph G as a random-order stream of edges, an ϵ-accurate oracle $\mathcal{O} \to \{-1, 1\}$, a high-degree threshold $\theta = \frac{\epsilon^2 m}{c}$ (where $c = \frac{80}{\delta}$).
Output: The estimate of the MAX-CUT value of G.

▷ **Preprocessing phase**
1: **initialize** $F \leftarrow \emptyset$, $\tilde{H} \leftarrow \emptyset$, $e(L^+, L^-) \leftarrow 0$.
▷ **Streaming phase**
2: **for each** edge (u, v) in the first $\frac{\beta}{\delta^3\epsilon^4}$ edges of the stream (where β is a sufficiently large universal constant) **do**
3: $F \leftarrow F \cup \{(u, v)\}$, $\tilde{H} \leftarrow \tilde{H} \cup \{u, v\}$
4: **for each** vertex $v \in \tilde{H}$ **do**
5: **initialize** $e(v, V \setminus \tilde{H}) \leftarrow 0$, $e(v, L^+) \leftarrow 0$, $e(v, L^-) \leftarrow 0$.
6: **for each** remaining edge (u, v) in the stream **do**
7: $Y_u = \mathcal{O}(u), Y_v = \mathcal{O}(v)$
8: **if** $u \in \tilde{H}$ and $v \in \tilde{H}$ **then** $F \leftarrow F \cup \{(u, v)\}$
9: **else**
10: **if** $u \in \tilde{H}$ and $v \in V \setminus \tilde{H}$ **then** $e(u, V \setminus \tilde{H}) \leftarrow e(u, V \setminus \tilde{H}) + 1$
11: **if** $v \in \tilde{H}$ and $u \in V \setminus \tilde{H}$ **then** $e(v, V \setminus \tilde{H}) \leftarrow e(v, V \setminus \tilde{H}) + 1$
12: **if** $u \in \tilde{H}$ and $v \in V \setminus \tilde{H}$ **then**
13: **if** $Y_v = 1$ **then** $e(u, L^+) \leftarrow e(u, L^+) + 1$ **else** $e(u, L^-) \leftarrow e(u, L^-) + 1$
14: **if** $u \in V \setminus \tilde{H}$ and $v \in V \setminus \tilde{H}$ and $Y_u \neq Y_v$ **then**
15: $e(L^+, L^-) \leftarrow e(L^+, L^-) + 1$ (i.e., apply Algorithm 1 on $G[V \setminus \tilde{H}]$)
▷ **Postprocessing phase**
16: $H \leftarrow \{v \in \tilde{H} : |\{e \in F : v \in e\}| + e(v, V \setminus \tilde{H}) \geq \theta\}$, $L \leftarrow V \setminus H$
17: **for each** edge $(u, v) \in F$ **do**
18: $Y_u = \mathcal{O}(u), Y_v = \mathcal{O}(v)$
19: **if** $u \in H$ and $v \in L$ **then**
20: **if** $Y_v = 1$ **then** $e(u, L^+) \leftarrow e(u, L^+) + 1$ **else** $e(u, L^-) \leftarrow e(u, L^-) + 1$
21: **if** $u \in \tilde{H} \setminus H$ and $v \in \tilde{H} \setminus H$ and $Y_u \neq Y_v$ **then** $e(L^+, L^-) \leftarrow e(L^+, L^-) + 1$
22: **for each** vertex $v \in \tilde{H} \setminus H$ **do**
23: $Y_v = \mathcal{O}(v)$
24: **if** $Y_v = 1$ **then** $e(L^+, L^-) \leftarrow e(L^+, L^-) + e(v, L^-)$ **else** $e(L^+, L^-) \leftarrow e(L^+, L^-) + e(v, L^+)$
25: $\mathrm{ALG}_1 \leftarrow e(L^+, L^-) + \sum_{v \in H} \max\{e(v, L^-), e(v, L^+)\}$
26: $\mathrm{ALG}_2 \leftarrow \sum_{v \in H}(e(v, L^-) + e(v, L^+))$
27: **return** $\max\{\mathrm{ALG}_1, \mathrm{ALG}_2\}$

Proof of Theorem 9. In Algorithm 2, we store the first $\frac{\beta}{\delta^3\epsilon^4}$ edges of the stream (Line 3) and the remaining edges with both endpoints in $\tilde{H}$ (Line 8). We also maintain several counters for vertices in $\tilde{H}$. Therefore, the total space complexity of Algorithm 2 is $O(\frac{1}{\delta^6\epsilon^8})$ words.

Next, we show the approximation guarantee of Algorithm 2. Note that at the end of the stream we can identify H exactly, using the information stored during the streaming phase. In Algorithm 2, we apply Algorithm 1 to the subgraph $G[V \setminus \tilde{H}]$ during the streaming phase (Line 15). Hypothetically, during the postprocessing phase, we apply Algorithm 1 to the subgraph $G[\tilde{H} \setminus H]$ (Line 21) and to edges with one endpoint in $\tilde{H} \setminus H$ and the

other endpoint in $V \setminus \tilde{H}$ (Line 24). This is equivalent to applying Algorithm 1 to $G[L]$ since $L = (V \setminus \tilde{H}) \cup (\tilde{H} \setminus H)$. Then the approximation guarantee of Algorithm 2 follows directly from Theorem 6 (with failure probability $\frac{\delta}{2}$) and Lemma 10 by using union bound. ◀

4.2 Arbitrary Order Streams

Overview of the Algorithm. Unlike random order streams, where we can identify a set $\tilde{H}$, containing the high-degree vertices H, by storing only a small number of edges from the beginning of the stream, finding high-degree vertices is more challenging in arbitrary order streams. To handle this, we employ *reservoir sampling* [47]. Specifically, we uniformly sample $\text{poly}(1/\epsilon, 1/\delta)$ edges from the stream. Then, at the end of the stream, we use these sampled edges to compute a small set $\tilde{H}$ such that $\tilde{H} \supseteq H$. This approach is similar to what we used in the random order stream; however, in this case, we can only retrieve $\tilde{H}$ rather than H at the end of the stream.

We also need to estimate the number of incident edges related to high-degree vertices. To this end, we use techniques from vector sketching, particularly those developed for *frequency estimation* and *heavy hitters*. We consider the sketching techniques for heavy hitters, specifically the randomized summaries of CountMin sketch [15], corresponding to V^+ and V^-, denoted by $\mathbf{CM}[V^+]$ and $\mathbf{CM}[V^-]$, respectively. Intuitively, by the end of the stream, $\mathbf{CM}[V^+]$ and $\mathbf{CM}[V^-]$ will contain the estimates of the number of incident edges of high-degree vertices to sets V^+ and V^-, respectively. More precisely, as an edge (u, v) arrive, we increment the counters related to u in $\mathbf{CM}[V^{\mathcal{O}(v)}]$, and increment the counters related to v in $\mathbf{CM}[V^{\mathcal{O}(u)}]$, where $\mathcal{O}(v)$ and $\mathcal{O}(u)$ are respectively the predicted sign of v, u by the given ϵ-accurate oracle $\mathcal{O}$.

Note that we cannot detect H exactly by the end of stream. Instead, we have $\tilde{H}$. We use $\mathbf{CM}[V^+]$ and $\mathbf{CM}[V^-]$ to approximately compute the value of the cut $(\tilde{L}^+, \tilde{L}^-)$ on the induced subgraph $G[\tilde{L}]$, where $\tilde{L} := V \setminus \tilde{H}$. Then, using $\mathbf{CM}[V^+]$ and $\mathbf{CM}[V^-]$, we can approximately run the greedy approach for $\tilde{H}$. Unlike the random order setting where we could implement the greedy extension exactly, here we can only store the approximate values. However, we show that our estimates of the number of incident edges of high-degree vertices to sets V^+ and V^- are only off by $\text{poly}(\epsilon, \delta) \cdot m$ additive terms with high probability, and they suffice to provide a strictly better than $\frac{1}{2}$-approximation for MAX-CUT on $G[\tilde{L}]$ and its extension with $\tilde{H}$. Similarly, we can use $\mathbf{CM}[V^+]$ and $\mathbf{CM}[V^-]$ to compute the size of $(\tilde{H}, \tilde{L})$ approximately too. Then, we can show that for sufficiently small value of ϵ, the best of two candidate cuts is a $(\frac{1}{2} + \Omega(\epsilon^2))$-approximation for MAX-CUT on G with probability $1 - \delta$.

Next, we prove the approximation guarantee and the space complexity of our algorithm for arbitrary order streams. The algorithm is formally given in Algorithm 3.

▶ **Theorem 11.** *Let $\epsilon \in (0, \frac{1}{2}]$ and $\delta \in (0, 1)$. Given an ϵ-accurate oracle $\mathcal{O}$ for the MAX-CUT of G, there exists a single-pass $(\frac{1}{2} + \Omega(\epsilon^2))$-approximation algorithm for estimating the MAX-CUT value of G in the insertion-only arbitrary-order streams. The algorithm uses $O(\frac{1}{\epsilon^7 \delta^3} \ln \frac{1}{\epsilon\delta})$ words of space. The approximation holds with probability at least $1 - \delta$.*

▶ **Lemma 12.** *Let $\delta \in (0, 1)$. Then $\tilde{H}$ contains all high-degree vertices H in G with probability at least $1 - \frac{\delta}{2}$.*

Proof. The proof is basically similar to that of Lemma 10. Suppose that v is a high-degree vertex in G, i.e., $\deg(v) \geq \frac{\epsilon^2 m}{c}$. For each edge e_i incident to v, define an indicator random variable

$$X_i = \begin{cases} 1, & \text{if } e_i \in F \\ 0, & \text{otherwise} \end{cases}$$

Algorithm 3 Estimating the MAX-CUT value in (insertion-only) arbitrary-order streams.

Input: Graph G as an arbitrary-order stream of edges, an ϵ-accurate oracle $\mathcal{O} \to \{-1, 1\}$
Output: Estimate of the MAX-CUT value of G

▷ **Preprocessing phase**

Initialize $e(V^+, V^-) \leftarrow 0$.
Set $h_1, \ldots, h_k : [n] \to [w]$ be 2-wise independent hash functions, with $k = \lceil \frac{e}{\epsilon^7 \delta^3} \rceil$ and $w = \lceil \ln \frac{8\beta}{\epsilon^4 \delta^4} \rceil$ (where β is a sufficiently large universal constant).
Initialize $\mathbf{CM}[V^+]$ and $\mathbf{CM}[V^-]$ to zero.

▷ **Streaming phase**

for each edge (u, v) in the stream **do**
    $Y_u = \mathcal{O}(u), Y_v = \mathcal{O}(v)$
    if $Y_u \neq Y_v$ **then** $e(V^+, V^-) \leftarrow e(V^+, V^-) + 1$
    for each $\ell \in [k]$ **do**
        $\mathbf{CM}[V^{\mathcal{O}(u)}][\ell, h_\ell(v)] \leftarrow \mathbf{CM}[V^{\mathcal{O}(u)}][\ell, h_\ell(v)] + 1$
        $\mathbf{CM}[V^{\mathcal{O}(v)}][\ell, h_\ell(u)] \leftarrow \mathbf{CM}[V^{\mathcal{O}(v)}][\ell, h_\ell(u)] + 1$
In parallel, uniformly sample $\frac{\beta}{\delta^3 \epsilon^4}$ edges F in the stream via reservoir sampling [47].

▷ **Postprocessing phase**

for each $v \in V$ **do**
    $f_v^+ \leftarrow \min_{\ell \in [k]} \mathbf{CM}[V^+][\ell, h_\ell(v)]$
    $f_v^- \leftarrow \min_{\ell \in [k]} \mathbf{CM}[V^-][\ell, h_\ell(v)]$
$\tilde{H} := \bigcup_{(u,v) \in F} \{u, v\}$
$\tilde{H}^+ \leftarrow \{v \in \tilde{H} : Y_v = 1\}, \tilde{H}^- \leftarrow V \setminus \tilde{H}^+$
$\tilde{e}(\tilde{L}^+, \tilde{L}^-) \leftarrow e(V^+, V^-) - \sum_{v \in \tilde{H}^+} f_v^- - \sum_{v \in \tilde{H}^-} f_v^+$
$\mathrm{ALG}_1 \leftarrow \tilde{e}(\tilde{L}^+, \tilde{L}^-) + \sum_{v \in \tilde{H}} \max\{f_v^-, f_v^+\}$
$\mathrm{ALG}_2 \leftarrow \sum_{v \in \tilde{H}} (f_v^- + f_v^+)$
return $\max\{\mathrm{ALG}_1, \mathrm{ALG}_2\}$

So, $\mathbb{E}[X_i] = \Pr[X_i = 1] = \frac{\beta}{\delta^3 \epsilon^4 m}$. We define $X := \sum_{i \in [\deg(v)]} X_i$ to denote the number of edges incident to v that are sampled by the end of the stream. Then, $\mathbb{E}[X] = \sum_{i \in [\deg(v)]} \mathbb{E}[X_i] = \frac{\beta \deg(v)}{\delta^3 \epsilon^4 m}$.

Next, we bound $\mathrm{Var}[X]$. For any $i \in [\deg(v)]$, $\mathrm{Var}[X_i] = \mathbb{E}[X_i^2] - (\mathbb{E}[X_i])^2 = \mathbb{E}[X_i] - (\mathbb{E}[X_i])^2 = \frac{\beta}{\delta^3 \epsilon^4 m} - (\frac{\beta}{\delta^3 \epsilon^4 m})^2 = \frac{\beta}{\delta^3 \epsilon^4 m} - \frac{\beta^2}{\delta^6 \epsilon^8 m^2}$. Since $(X_i)_{i \in [\deg(v)]}$ are negatively correlated, $\mathrm{Var}[X] \leq \sum_{i \in [\deg(v)]} \mathrm{Var}[X_i] = \deg(v) \cdot \left(\frac{\beta}{\delta^3 \epsilon^4 m} - \frac{\beta^2}{\delta^6 \epsilon^8 m^2} \right) \leq \frac{\beta \deg(v)}{\delta^3 \epsilon^4 m}$. Therefore, by Chebyshev's inequality,

$$\begin{aligned}
\Pr[v \notin \tilde{H}] = \Pr[X \leq 0] &= \Pr\left[X \leq \mathbb{E}[X] - \frac{\beta \deg(v)}{\delta^3 \epsilon^4 m}\right] \\
&\leq \frac{\mathrm{Var}[X] \cdot \delta^6 \epsilon^8 m^2}{\beta^2 \deg^2(v)} \\
&\leq \frac{\delta^3 \epsilon^4 m}{\beta \cdot \deg(v)} && \triangleright\ \mathrm{Var}[X] \leq \frac{\beta \deg(v)}{\delta^3 \epsilon^4 m} \\
&\leq \frac{\delta^3 \epsilon^2 c}{\beta}. && \triangleright\ \deg(v) \geq \frac{\epsilon^2 m}{c}
\end{aligned}$$

By our definition of high-degree vertices, there are at most $\frac{2m}{\epsilon^2 m/c} = \frac{2c}{\epsilon^2}$ high-degree vertices in G. By union bound, with probability at least $1 - \frac{2c}{\epsilon^2} \cdot \frac{\delta^3 \epsilon^2 c}{\beta} = 1 - \frac{2c^2 \delta^3}{\beta} \geq 1 - \frac{\delta}{2}$ (since $c = \frac{80}{\delta}$ and β sufficiently large), $\tilde{H}$ contains all high-degree vertices in G. ◀

▶ **Lemma 13.** *Let $\epsilon \in (0, \frac{1}{2}]$ and $\delta \in (0,1)$. For all vertices $v \in \tilde{H}$, (1) $f_v^+ \geq e(v, V^+)$ and $f_v^- \geq e(v, V^-)$. (2) with probability at least $1 - \frac{\delta}{4}$, $f_v^+ \leq e(v, V^+) + 2\epsilon^7\delta^3 m$ and $f_v^- \leq e(v, V^-) + 2\epsilon^7\delta^3 m$.*

Proof. (1) By Theorem 2, we have $f_v^+ \geq e(v, V^+)$ and $f_v^- \geq e(v, V^-)$ for all $v \in V$.

(2) By Theorem 2, we have $f_v^+ \leq e(v, V^+) + 2\epsilon^7\delta^3 m$ and $f_v^- \leq e(v, V^-) + 2\epsilon^7\delta^3 m$ with probability at least $1 - \frac{\epsilon^4\delta^4}{8\beta}$ for any $v \in V$. Since $|\tilde{H}| \leq \frac{2\beta}{\delta^3\epsilon^4}$, by union bound, with probability at least $1 - \frac{\epsilon^4\delta^4}{8\beta} \cdot \frac{2\beta}{\delta^3\epsilon^4} = 1 - \frac{\delta}{4}$, we have $f_v^+ \leq e(v, V^+) + 2\epsilon^7\delta^3 m$ and $f_v^- \leq e(v, V^-) + 2\epsilon^7\delta^3 m$ for all $v \in \tilde{H}$. ◀

Proof of Theorem 11. In Algorithm 3, we use CountMin sketch with $k = \lceil \frac{e}{\epsilon^7\delta^3} \rceil$ and $w = \lceil \ln \frac{8\beta}{\epsilon^4\delta^4} \rceil$ to estimate the number of incident edges of high-degree vertices to with respect to V^+ and V^-, which takes $O(\frac{1}{\epsilon^7\delta^3} \ln \frac{1}{\epsilon^4\delta^4}) = O(\frac{1}{\epsilon^7\delta^3} \ln \frac{1}{\epsilon\delta})$ words of space, by Theorem 2. Also, we use reservoir sampling to uniformly sample $O(\frac{1}{\delta^3\epsilon^4})$ edges, which takes $O(\frac{1}{\delta^3\epsilon^4})$ words of space. Therefore, the total space complexity of Algorithm 3 is $O(\frac{1}{\epsilon^7\delta^3} \ln \frac{1}{\epsilon\delta})$ words.

Next, we show the approximation guarantee of Algorithm 3. Recall that in arbitrary order streams, we cannot detect H by the end of stream, and we have $\tilde{H}$ instead. Let $A_1 := e(\tilde{L}^+, \tilde{L}^-) + \sum_{v \in \tilde{H}} \max\{e(v, \tilde{L}^-), e(v, \tilde{L}^+)\}$ denote the value of the cut $(\tilde{C}, V \setminus \tilde{C})$ obtained by running Algorithm 1 on $G[\tilde{L}]$ and then assigning the vertices in $\tilde{H}$ in a greedy manner. Let $A_2 := \sum_{v \in \tilde{H}} (e(v, \tilde{L}^-) + e(v, \tilde{L}^+))$ denote the value of the cut $(\tilde{H}, \tilde{L})$. Suppose that the best of $(\tilde{C}, V \setminus \tilde{C})$ and $(\tilde{H}, \tilde{L})$ cuts is a $(\frac{1}{2} + \frac{\epsilon^2}{16})$-approximation for the MAX-CUT value of G. Recall that $\mathrm{ALG}_1 = \tilde{e}(\tilde{L}^+, \tilde{L}^-) + \sum_{v \in \tilde{H}} \max\{f_v^-, f_v^+\}$ and $\mathrm{ALG}_2 = \sum_{v \in \tilde{H}} (f_v^- + f_v^+)$. In the following, we show that ALG_1 and ALG_2 are good approximations of A_1 and A_2, respectively. Therefore, $\max\{\mathrm{ALG}_1, \mathrm{ALG}_2\}$ returned by Algorithm 3 is also a strictly better than $\frac{1}{2}$-approximation for the MAX-CUT value of G.

Since $V^+ = \tilde{H}^+ \cup \tilde{L}^+$ and $V^- = \tilde{H}^- \cup \tilde{L}^-$, we have $e(v, V^+) = e(v, \tilde{H}^+) + e(v, \tilde{L}^+)$ and $e(v, V^-) = e(v, \tilde{H}^-) + e(v, \tilde{L}^-)$ for any vertex $v \in V$. By Lemma 13, with probability at least $1 - \frac{\delta}{4}$, for all vertices $v \in \tilde{H}$, we have

$$e(v, \tilde{H}^+) + e(v, \tilde{L}^+) \leq f_v^+ \leq e(v, \tilde{H}^+) + e(v, \tilde{L}^+) + 2\epsilon^7\delta^3 m,$$
$$e(v, \tilde{H}^-) + e(v, \tilde{L}^-) \leq f_v^- \leq e(v, \tilde{H}^-) + e(v, \tilde{L}^-) + 2\epsilon^7\delta^3 m.$$

Since $|\tilde{H}| \leq \frac{2\beta}{\delta^3\epsilon^4}$, we have $0 \leq e(v, \tilde{H}^+), e(v, \tilde{H}^-) \leq \frac{2\beta}{\delta^3\epsilon^4}$. Therefore, we have

$$e(v, \tilde{L}^+) \leq f_v^+ \leq \frac{2\beta}{\delta^3\epsilon^4} + 2\epsilon^7\delta^3 m + e(v, \tilde{L}^+),$$
$$e(v, \tilde{L}^-) \leq f_v^- \leq \frac{2\beta}{\delta^3\epsilon^4} + 2\epsilon^7\delta^3 m + e(v, \tilde{L}^-).$$

Then we have

$$0 \leq f_v^+ - e(v, \tilde{L}^+) \leq \frac{2\beta}{\delta^3\epsilon^4} + 2\epsilon^7\delta^3 m \quad \text{and} \quad 0 \leq f_v^- - e(v, \tilde{L}^-) \leq \frac{2\beta}{\delta^3\epsilon^4} + 2\epsilon^7\delta^3 m,$$

and

$$\max\{e(v, \tilde{L}^+), e(v, \tilde{L}^-)\} \leq \max\{f_v^+, f_v^-\} \leq \frac{2\beta}{\delta^3\epsilon^4} + 2\epsilon^7\delta^3 m + \max\{e(v, \tilde{L}^+), e(v, \tilde{L}^-)\}.$$

Note that

$$e(\tilde{L}^+, \tilde{L}^-) = e(V^+, V^-) - e(\tilde{H}^+, \tilde{H}^-) - \sum_{v \in \tilde{H}^+} e(v, \tilde{L}^-) - \sum_{v \in \tilde{H}^-} e(v, \tilde{L}^+),$$
$$\tilde{e}(\tilde{L}^+, \tilde{L}^-) = e(V^+, V^-) - \sum_{v \in \tilde{H}^+} f_v^- - \sum_{v \in \tilde{H}^-} f_v^+.$$

We have

$$\begin{aligned} e(\tilde{L}^+, \tilde{L}^-) - \tilde{e}(\tilde{L}^+, \tilde{L}^-) &= \sum_{v \in \tilde{H}^+} (f_v^- - e(v, \tilde{L}^-)) + \sum_{v \in \tilde{H}^-} (f_v^+ - e(v, \tilde{L}^+)) - e(\tilde{H}^+, \tilde{H}^-) \\ &\leq \frac{2\beta}{\delta^3\epsilon^4} \cdot \left(\frac{2\beta}{\delta^3\epsilon^4} + 2\epsilon^7\delta^3 m \right) + \frac{2\beta}{\delta^3\epsilon^4} \cdot \left(\frac{2\beta}{\delta^3\epsilon^4} + 2\epsilon^7\delta^3 m \right) - 0 \\ &= \frac{8\beta^2}{\delta^6\epsilon^8} + 8\beta\epsilon^3 m. \end{aligned}$$

Therefore,

$$\begin{aligned} A_1 - \mathrm{ALG}_1 &= (e(\tilde{L}^+, \tilde{L}^-) - \tilde{e}(\tilde{L}^+, \tilde{L}^-)) + \sum_{v \in \tilde{H}} (\max\{e(v, \tilde{L}^-), e(v, \tilde{L}^+)\} - \max\{f_v^-, f_v^+\}) \\ &\leq \left(\frac{8\beta^2}{\delta^6\epsilon^8} + 8\beta\epsilon^3 m \right) + \frac{2\beta}{\delta^3\epsilon^4} \cdot 0 = \frac{8\beta^2}{\delta^6\epsilon^8} + 8\beta\epsilon^3 m. \end{aligned}$$

So, $\mathrm{ALG}_1 \geq A_1 - (\frac{8\beta^2}{\delta^6\epsilon^8} + 8\beta\epsilon^3 m) = A_1 - \Theta(\epsilon^3 m)$.

Similarly,

$$A_2 - \mathrm{ALG}_2 = \sum_{v \in \tilde{H}} ((e(v, \tilde{L}^-) - f_v^-) + (e(v, \tilde{L}^+) - f_v^+)) \leq \frac{2\beta}{\delta^3\epsilon^4} \cdot (0 + 0) = 0.$$

So, $\mathrm{ALG}_2 \geq A_2$.

Since we assume that $\max\{A_1, A_2\}$ is a $(\frac{1}{2} + \frac{\epsilon^2}{16})$-approximation for the MAX-CUT value of G, we have $\mathrm{ALG} := \max\{\mathrm{ALG}_1, \mathrm{ALG}_2\} \geq \max\{A_1, A_2\} - \Theta(\epsilon^3 m) \geq (\frac{1}{2} + \frac{\epsilon^2}{16}) \cdot \mathrm{OPT} - \Theta(\epsilon^3 \cdot \mathrm{OPT}) = (\frac{1}{2} + \Omega(\epsilon^2)) \cdot \mathrm{OPT}$.

Finally, it remains to show that the best of $(\tilde{C}, V \setminus \tilde{C})$ and $(\tilde{H}, \tilde{L})$ cuts is a $(\frac{1}{2} + \frac{\epsilon^2}{16})$-approximation for the MAX-CUT value of G. This directly follows from Theorem 6 (with failure probability $\frac{\delta}{4}$), by substituting H and L with $\tilde{H}$ and $\tilde{L}$, respectively. Together with Lemma 12, Lemma 13, and applying union bound, this concludes the proof. ◀

4.3 Extension to Dynamic Streams

Overview of the Algorithm. Our algorithm in dynamic streams is basically similar to Algorithm 3. The only difference is that to compute a small set $\tilde{H}$ at the end of the stream such that $\tilde{H} \supseteq H$, instead of reservoir sampling, we need to use ℓ_0-sampling [28].

Next, we prove the approximation guarantee and the space complexity of our algorithm for dynamic streams. The algorithm is formally given in Algorithm 4.

▶ **Theorem 14.** *Let $\epsilon \in (0, \frac{1}{2}]$ and $\delta \in (0, 1)$. Given an ϵ-accurate oracle $\mathcal{O}$ for the MAX-CUT of G, there exists a single-pass $(\frac{1}{2} + \Omega(\epsilon^2))$-approximation algorithm for estimating the MAX-CUT value of G in the dynamic streams. The algorithm uses $O(\frac{\log^2 n}{\epsilon^7\delta^3} \log \frac{1}{\epsilon\delta})$ bits of space. The approximation holds with probability at least $1 - \delta$.*

Note that Theorem 1 follows from Theorem 11 and Theorem 14 by setting $\delta = 1/3$.

Proof Sketch of Theorem 14. The proof follows a similar structure to that of Theorem 11. First, we show that ℓ_0-sampling in dynamic streams can effectively replace reservoir sampling in insertion-only streams, allowing us to retrieve a set $\tilde{H} \supseteq H$ with high probability by the end of the stream.

Algorithm 4 Estimating the MAX-CUT value in dynamic streams.

Input: Graph G as a dynamic stream of edges, an ϵ-accurate oracle $\mathcal{O} \to \{-1, 1\}$
Output: Estimate of the MAX-CUT value of G

▷ **Preprocessing phase**

Initialize $e(V^+, V^-) \leftarrow 0$.
Set $h_1, \ldots, h_k : [n] \to [w]$ be 2-wise independent hash functions, with $k = \lceil \frac{e}{\epsilon^7 \delta^3} \rceil$ and $w = \lceil \ln \frac{8\beta}{\epsilon^4 \delta^4} \rceil$ (where β is a sufficiently large universal constant).
Initialize $\mathbf{CM}[V^+]$ and $\mathbf{CM}[V^-]$ to zero.

▷ **Streaming phase**

for each item $((u, v), \Delta_{(u,v)})$ in the stream **do**
    $Y_u = \mathcal{O}(u), Y_v = \mathcal{O}(v)$
    if $Y_u \neq Y_v$ **then** $e(V^+, V^-) \leftarrow e(V^+, V^-) + 1$
    for each $\ell \in [k]$ **do**
        $\mathbf{CM}[V^{\mathcal{O}(u)}][\ell, h_\ell(v)] \leftarrow \mathbf{CM}[V^{\mathcal{O}(u)}][\ell, h_\ell(v)] + \Delta_{(u,v)}$
        $\mathbf{CM}[V^{\mathcal{O}(v)}][\ell, h_\ell(u)] \leftarrow \mathbf{CM}[V^{\mathcal{O}(v)}][\ell, h_\ell(u)] + \Delta_{(u,v)}$
In parallel, maintain $\frac{\beta}{\delta^3 \epsilon^4}$ ℓ_0-samplers (with failure probability $\delta' = \frac{\delta^4 \epsilon^4}{8\beta}$) to obtain $\frac{\beta}{\delta^3 \epsilon^4}$ different edges F.

▷ **Postprocessing phase**

for each $v \in V$ **do**
    $f_v^+ \leftarrow \min_{\ell \in [k]} \mathbf{CM}[V^+][\ell, h_\ell(v)]$
    $f_v^- \leftarrow \min_{\ell \in [k]} \mathbf{CM}[V^-][\ell, h_\ell(v)]$
$\tilde{H} := \bigcup_{(u,v) \in F} \{u, v\}$
$\tilde{H}^+ \leftarrow \{v \in \tilde{H} : Y_v = 1\}, \tilde{H}^- \leftarrow V \setminus \tilde{H}^+$
$\tilde{e}(\tilde{L}^+, \tilde{L}^-) \leftarrow e(V^+, V^-) - \sum_{v \in \tilde{H}^+} f_v^- - \sum_{v \in \tilde{H}^-} f_v^+$
$\mathrm{ALG}_1 \leftarrow \tilde{e}(\tilde{L}^+, \tilde{L}^-) + \sum_{v \in \tilde{H}} \max\{f_v^-, f_v^+\}$
$\mathrm{ALG}_2 \leftarrow \sum_{v \in \tilde{H}} (f_v^- + f_v^+)$
return $\max\{\mathrm{ALG}_1, \mathrm{ALG}_2\}$

Next, we show that for all vertices $v \in \tilde{H}$, our estimates of the number of incident edges from v to the sets V^+ and V^- has additive error at most $\mathrm{poly}(\epsilon, \delta) \cdot m$, with high probability.

Finally, using arguments similar to those in the proof of Theorem 11, we can show that Algorithm 4 provides a $(\frac{1}{2} + \Omega(\epsilon^2))$-approximation for the MAX-CUT value of G with probability $1 - \delta$. For completeness, we provide the detailed proof in Section A.

5 Conclusion

We present the first learning-augmented algorithm for estimating the value of MAX-CUT in the streaming setting by leveraging ϵ-accurate predictions. Specifically, we provide a single-pass streaming algorithm that achieves a $(1/2 + \Omega(\epsilon^2))$-approximation for estimating the MAX-CUT value of a graph in insertion-only (respectively, fully dynamic) streams using $\mathrm{poly}(1/\epsilon)$ (respectively, $\mathrm{poly}(1/\epsilon, \log n)$) words of space. This result contrasts with the lower bound in the classical streaming setting (without predictions), where any (randomized) single-pass streaming algorithm that achieves an approximation ratio of at least $1/2 + \epsilon$ requires $\Omega(n/2^{\mathrm{poly}(1/\epsilon)})$ space.

Our work leaves several questions for further research. For example, it would be interesting to extend our algorithm from unweighted to weighted graphs. Currently, our algorithm for handling the low-degree part does not seem to work for weighted graphs, as an edge with a

heavy weight but incorrectly predicted signs can significantly affect our estimator. Another open question is to establish lower bounds on the trade-offs between approximation ratio and space complexity when the streaming algorithm is provided with predictions.

References

1. Anders Aamand, Justin Chen, Huy Nguyen, Sandeep Silwal, and Ali Vakilian. Improved frequency estimation algorithms with and without predictions. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
2. Anders Aamand, Piotr Indyk, and Ali Vakilian. frequency estimation algorithms under zipfian distribution. *arXiv preprint*, 2019. arXiv:1908.05198.
3. Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Graph sketches: sparsification, spanners, and subgraphs. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on Principles of Database Systems (PODS)*, pages 5–14, 2012. doi:10.1145/2213556.2213560.
4. Spyros Angelopoulos, Christoph Dürr, Shendan Jin, Shahin Kamali, and Marc Renault. Online computation with untrusted advice. In *11th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 151 of *LIPIcs*, pages 52:1–52:15, 2020. doi:10.4230/LIPICS.ITCS.2020.52.
5. Antonios Antoniadis, Christian Coester, Marek Eliás, Adam Polak, and Bertrand Simon. Online metric algorithms with untrusted predictions. *ACM Trans. Algorithms*, 19(2):19:1–19:34, 2023. doi:10.1145/3582689.
6. Antonios Antoniadis, Themis Gouleakis, Pieter Kleer, and Pavel Kolev. Secretary and online matching problems with machine learned advice. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.
7. Etienne Bamas, Andreas Maggiori, and Ola Svensson. The primal-dual method for learning augmented algorithms. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.
8. Siddhartha Banerjee, Vincent Cohen-Addad, Anupam Gupta, and Zhouzi Li. Graph searching with predictions. In *14th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 251 of *LIPIcs*, pages 12:1–12:24, 2023. doi:10.4230/LIPICS.ITCS.2023.12.
9. Arnab Bhattacharyya and Yuichi Yoshida. *Property Testing - Problems and Techniques*. Springer, 2022. doi:10.1007/978-981-16-8622-1.
10. Jan van den Brand, Sebastian Forster, Yasamin Nazari, and Adam Polak. On dynamic graph algorithms with predictions. In *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3534–3557, 2024. doi:10.1137/1.9781611977912.126.
11. Vladimir Braverman, Prathamesh Dharangutte, Vihan Shah, and Chen Wang. Learning-augmented maximum independent set. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, volume 317 of *LIPIcs*, pages 24:1–24:18, 2024. doi:10.4230/LIPICS.APPROX/RANDOM.2024.24.
12. Justin Chen, Sandeep Silwal, Ali Vakilian, and Fred Zhang. Faster fundamental graph algorithms via learned predictions. In *International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 3583–3602, 2022. URL: https://proceedings.mlr.press/v162/chen22v.html.
13. Justin Y. Chen, Talya Eden, Piotr Indyk, Honghao Lin, Shyam Narayanan, Ronitt Rubinfeld, Sandeep Silwal, Tal Wagner, David P. Woodruff, and Michael Zhang. Triangle and four cycle counting with predictions in graph streams. In *The Tenth International Conference on Learning Representations (ICLR)*, 2022. URL: https://openreview.net/forum?id=8in_5gN9I0.
14. Vincent Cohen-Addad, Tommaso d'Orsi, Anupam Gupta, Euiwoong Lee, and Debmalya Panigrahi. Learning-augmented approximation algorithms for maximum cut and related problems. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

15 Graham Cormode and Shan Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005. doi:10.1016/J.JALGOR.2003.12.001.

16 Sami Davies, Benjamin Moseley, Sergei Vassilvitskii, and Yuyan Wang. Predictive flows for faster ford-fulkerson. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 7231–7248, 2023. URL: https://proceedings.mlr.press/v202/davies23b.html.

17 Adela F DePavia, Erasmo Tani, and Ali Vakilian. Learning-based algorithms for graph searching problems. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 238 of *Proceedings of Machine Learning Research*, pages 928–936, 2024. URL: https://proceedings.mlr.press/v238/depavia24a.html.

18 Michael Dinitz, Sungjin Im, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. Faster matchings via learned duals. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 10393–10406, 2021. URL: https://proceedings.neurips.cc/paper/2021/hash/5616060fb8ae85d93f334e7267307664-Abstract.html.

19 Talya Eden, Piotr Indyk, Shyam Narayanan, Ronitt Rubinfeld, Sandeep Silwal, and Tal Wagner. Learning-based support estimation in sublinear time. In *9th International Conference on Learning Representations (ICLR)*, 2021. URL: https://openreview.net/forum?id=tilovEHA3YS.

20 Paolo Ferragina and Giorgio Vinciguerra. The pgm-index: a fully-dynamic compressed learned index with provable worst-case bounds. *Proc. VLDB Endow.*, 13(8):1162–1175, 2020. doi:10.14778/3389133.3389135.

21 Suprovat Ghoshal, Konstantin Makarychev, and Yury Makarychev. Constraint satisfaction problems with advice. In *Proceedings of the 2025 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2025.

22 Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995. doi:10.1145/227683.227684.

23 Monika Henzinger, Barna Saha, Martin P Seybold, and Christopher Ye. On the complexity of algorithms with predictions for dynamic graph problems. In *15th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 287 of *LIPIcs*, pages 62:1–62:25, 2024. doi:10.4230/LIPICS.ITCS.2024.62.

24 Chen-Yu Hsu, Piotr Indyk, Dina Katabi, and Ali Vakilian. Learning-based frequency estimation algorithms. In *7th International Conference on Learning Representations (ICLR)*, 2019.

25 Sungjin Im, Ravi Kumar, Mahshid Montazer Qaem, and Manish Purohit. Online knapsack with frequency predictions. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 2733–2743, 2021. URL: https://proceedings.neurips.cc/paper/2021/hash/161c5c5ad51fcc884157890511b3c8b0-Abstract.html.

26 Piotr Indyk, Ali Vakilian, and Yang Yuan. Learning-based low-rank approximations. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 7400–7410, 2019. URL: https://proceedings.neurips.cc/paper/2019/hash/1625abb8e458a79765c62009235e9d5b-Abstract.html.

27 Tanqiu Jiang, Yi Li, Honghao Lin, Yisong Ruan, and David P Woodruff. Learning-augmented data stream algorithms. In *8th International Conference on Learning Representations (ICLR)*, 2020.

28 Hossein Jowhari, Mert Sağlam, and Gábor Tardos. Tight bounds for lp samplers, finding duplicates in streams, and related problems. In *Proceedings of the 30th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS)*, pages 49–58, 2011.

29 Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Streaming lower bounds for approximating max-cut. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1263–1282, 2015. doi:10.1137/1.9781611973730.84.

30 Michael Kapralov, Sanjeev Khanna, Madhu Sudan, and Ameya Velingker. (1+ $\omega(1)$)-approximation to max-cut requires linear space. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1703–1722, 2017. doi:10.1137/1.9781611974782.112.

31 Michael Kapralov and Dmitry Krachun. An optimal space lower bound for approximating MAX-CUT. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 277–288, 2019. doi:10.1145/3313276.3316364.

32 Michael Kapralov, Aida Mousavifar, Cameron Musco, Christopher Musco, Navid Nouri, Aaron Sidford, and Jakab Tardos. Fast and space efficient spectral sparsification in dynamic streams. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1814–1833, 2020. doi:10.1137/1.9781611975994.111.

33 Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for max-cut and other 2-variable csps? *SIAM Journal on Computing*, 37(1):319–357, 2007. doi:10.1137/S0097539705447372.

34 Dmitry Kogan and Robert Krauthgamer. Sketching cuts in graphs and hypergraphs. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 367–376, 2015. doi:10.1145/2688073.2688093.

35 Ravi Kumar, Manish Purohit, and Zoya Svitkina. Improving online algorithms via ml predictions. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 9684–9693, 2018. URL: https://proceedings.neurips.cc/paper/2018/hash/73a427badebe0e32caa2e1fc7530b7f3-Abstract.html.

36 Silvio Lattanzi, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. Online scheduling via learned weights. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1859–1877, 2020. doi:10.1137/1.9781611975994.114.

37 Silvio Lattanzi, Ola Svensson, and Sergei Vassilvitskii. Speeding up bellman ford via minimum violation permutations. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 18584–18598, 2023. URL: https://proceedings.mlr.press/v202/lattanzi23a.html.

38 Yi Li, Honghao Lin, Simin Liu, Ali Vakilian, and David P. Woodruff. Learning the positions in countsketch. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

39 Honghao Lin, Tian Luo, and David P. Woodruff. Learning augmented binary search trees. In *International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 13431–13440, 2022. URL: https://proceedings.mlr.press/v162/lin22f.html.

40 Quanquan C Liu and Vaidehi Srinivas. The predicted-deletion dynamic model: Taking advantage of ml predictions, for free. *arXiv preprint*, 2023. doi:10.48550/arXiv.2307.08890.

41 Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. *J. ACM*, 68(4):24:1–24:25, 2021. doi:10.1145/3447579.

42 Mohammad Mahdian, Hamid Nazerzadeh, and Amin Saberi. Allocating online advertisement space with unreliable estimates. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 288–294, 2007. doi:10.1145/1250910.1250952.

43 Michael Mitzenmacher. A model for learned bloom filters and optimizing by sandwiching. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 462–471, 2018. URL: https://proceedings.neurips.cc/paper/2018/hash/0f49c89d1e7298bb9930789c8ed59d48-Abstract.html.

44 Atsuki Sato and Yusuke Matsui. Fast partitioned learned bloom filter. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

45 Nicholas Schiefer, Justin Y Chen, Piotr Indyk, Shyam Narayanan, Sandeep Silwal, and Tal Wagner. Learned interpolation for better streaming quantile approximation with worst-case guarantees. In *SIAM Conference on Applied and Computational Discrete Algorithms (ACDA)*, pages 87–97, 2023. doi:10.1137/1.9781611977714.8.

46 Kapil Vaidya, Eric Knorr, Michael Mitzenmacher, and Tim Kraska. Partitioned learned bloom filters. In *9th International Conference on Learning Representations (ICLR)*, 2021.

47 Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985. doi:10.1145/3147.3165.

A Proof of Theorem 14

▶ **Lemma 15.** *Let $\epsilon \in (0, \frac{1}{2}]$ and $\delta \in (0,1)$. With probability at least $1 - \frac{\delta}{4}$, Line 10 of Algorithm 4 returns a set F which contains $\frac{\beta}{\delta^3\epsilon^4}$ different edges.*

Proof. The error is introduced by the ℓ_0-samplers and the probability that sampling $\frac{\beta}{\delta^3\epsilon^4}$ uniform random edges in parallel does not yield $\frac{\beta}{\delta^3\epsilon^4}$ different edges.

By Theorem 4, the ℓ_0-sampler fails with probability at most δ'. Since we maintain $\frac{\beta}{\delta^3\epsilon^4}$ ℓ_0-samplers, by union bound, the failure probability introduced by ℓ_0-samplers is at most $\frac{\beta}{\delta^3\epsilon^4} \cdot \delta'$.

Suppose that all the ℓ_0-samplers succeed, i.e., $|F| = \frac{\beta}{\delta^3\epsilon^4}$. Now we bound the probability that there exist two identical edges in F. Consider two fixed edges $e_1, e_2 \in F$. Then $\Pr[e_1 = e_2] = \frac{1}{m}$. By union bound, the probability that there exists at least one pair of identical edges is at most $\binom{\beta/\delta^3\epsilon^4}{2} \cdot \frac{1}{m} \le \frac{\beta^2}{\delta^6\epsilon^8 m}$.

Therefore, the probability that F contains $\frac{\beta}{\delta^3\epsilon^4}$ different edges is at least $1 - \frac{\beta}{\delta^3\epsilon^4} \cdot \delta' - \frac{\beta^2}{\delta^6\epsilon^8 m} \ge 1 - \frac{\delta}{8} - \frac{\delta}{8} \ge 1 - \frac{\delta}{4}$ (since $m = \Omega(\epsilon^{-11}\delta^{-7})$). ◀

▶ **Lemma 16.** *Let $\delta \in (0,1)$. By the end of the stream, $\tilde{H}$ contains all high-degree vertices in G with probability at least $1 - \frac{\delta}{4}$.*

Proof. Suppose that v is a high-degree vertex in G, i.e., $\deg(v) \ge \frac{\epsilon^2 m}{c}$. For each edge e_i incident to v, define an indicator random variable

$$X_i = \begin{cases} 1, & \text{if } e_i \in F \\ 0, & \text{otherwise} \end{cases}$$

So, $\mathbb{E}[X_i] = \Pr[X_i = 1] = 1 - (1 - \frac{1}{m})^{\beta/\delta^3\epsilon^4} \ge 1 - \exp\left(-\frac{\beta}{\delta^3\epsilon^4 m}\right) \ge \frac{\beta}{\delta^3\epsilon^4 m} - \frac{\beta^2}{2\delta^6\epsilon^8 m^2} \ge \frac{\beta}{5\delta^3\epsilon^4 m}$ for sufficiently large m. We define $X := \sum_{i\in[\deg(v)]} X_i$ to denote the number of edges incident to v that are sampled by the end of the stream. Then, $\mathbb{E}[X] = \sum_{i\in[\deg(v)]} \mathbb{E}[X_i] \ge \frac{\beta \deg(v)}{5\delta^3\epsilon^4 m}$.

Next, we bound $\mathrm{Var}[X]$. For any $i \in [\deg(v)]$, $\mathrm{Var}[X_i] = \mathbb{E}[X_i^2] - (\mathbb{E}[X_i])^2 = \mathbb{E}[X_i] - (\mathbb{E}[X_i])^2 \le \frac{\beta}{5\delta^3\epsilon^4 m} - (\frac{\beta}{5\delta^3\epsilon^4 m})^2 = \frac{\beta}{5\delta^3\epsilon^4 m} - \frac{\beta^2}{25\delta^6\epsilon^8 m^2}$. Since $(X_i)_{i\in[\deg(v)]}$ are negatively correlated, $\mathrm{Var}[X] \le \sum_{i\in[\deg(v)]} \mathrm{Var}[X_i] = \deg(v) \cdot \left(\frac{\beta}{5\delta^3\epsilon^4 m} - \frac{\beta^2}{25\delta^6\epsilon^8 m^2}\right) \le \frac{\beta \deg(v)}{5\delta^3\epsilon^4 m}$. Therefore, by Chebyshev's inequality,

$$\begin{aligned}\Pr[v \notin \tilde{H}] = \Pr[X \le 0] &\le \Pr\left[X \le \mathbb{E}[X] - \frac{\beta \deg(v)}{5\delta^3\epsilon^4 m}\right] \\ &\le \frac{\mathrm{Var}[X] \cdot 25\delta^6\epsilon^8 m^2}{\beta^2 \deg^2(v)} \\ &\le \frac{5\delta^3\epsilon^4 m}{\beta \cdot \deg(v)} && \triangleright\ \mathrm{Var}[X] \le \frac{\beta \deg(v)}{5\delta^3\epsilon^4 m} \\ &\le \frac{5\delta^3\epsilon^2 c}{\beta}. && \triangleright\ \deg(v) \ge \frac{\epsilon^2 m}{c}\end{aligned}$$

By our definition of high-degree vertices, there are at most $\frac{2m}{\epsilon^2 m/c} = \frac{2c}{\epsilon^2}$ high-degree vertices in G. By union bound, with probability at least $1 - \frac{2c}{\epsilon^2} \cdot \frac{5\delta^3\epsilon^2 c}{\beta} = 1 - \frac{10c^2\delta^3}{\beta} \ge 1 - \frac{\delta}{4}$ (since $c = \frac{80}{\delta}$ and β sufficiently large), $\tilde{H}$ contains all high-degree vertices in G. ◀

▶ **Lemma 17.** *Let $\epsilon \in (0, \frac{1}{2}]$ and $\delta \in (0, 1)$. For all vertices $v \in \tilde{H}$, (1) $f_v^+ \ge e(v, V^+)$ and $f_v^- \ge e(v, V^-)$. (2) with probability at least $1 - \frac{\delta}{4}$, $f_v^+ \le e(v, V^+) + 2\epsilon^7\delta^3 m$ and $f_v^- \le e(v, V^-) + 2\epsilon^7\delta^3 m$.*

Proof. (1) By Theorem 2, we have $f_v^+ \ge e(v, V^+)$ and $f_v^- \ge e(v, V^-)$ for all $v \in V$.

(2) By Theorem 2, we have $f_v^+ \le e(v, V^+) + 2\epsilon^7\delta^3 m$ and $f_v^- \le e(v, V^-) + 2\epsilon^7\delta^3 m$ with probability at least $1 - \frac{\epsilon^4\delta^4}{8\beta}$ for any $v \in V$. Since $|\tilde{H}| \le \frac{2\beta}{\delta^3\epsilon^4}$, by union bound, with probability at least $1 - \frac{\epsilon^4\delta^4}{8\beta} \cdot \frac{2\beta}{\delta^3\epsilon^4} = 1 - \frac{\delta}{4}$, we have $f_v^+ \le e(v, V^+) + 2\epsilon^7\delta^3 m$ and $f_v^- \le e(v, V^-) + 2\epsilon^7\delta^3 m$ for all $v \in \tilde{H}$. ◀

Proof of Theorem 14. In Algorithm 4, we use CountMin sketch with $k = \lceil \frac{e}{\epsilon^7\delta^3} \rceil$ and $w = \lceil \ln \frac{8\beta}{\epsilon^4\delta^4} \rceil$ to estimate the number of incident edges of high-degree vertices to with respect to V^+ and V^-, which takes $O(\frac{1}{\epsilon^7\delta^3} \ln \frac{1}{\epsilon^4\delta^4}) = O(\frac{1}{\epsilon^7\delta^3} \ln \frac{1}{\epsilon\delta})$ words of space, by Theorem 2. Also, we maintain $O(\frac{1}{\delta^3\epsilon^4})$ ℓ_0-samplers, which takes $O(\frac{\log^2 n}{\delta^3\epsilon^4} \log \frac{1}{\delta^4\epsilon^4}) = O(\frac{\log^2 n}{\delta^3\epsilon^4} \log \frac{1}{\delta\epsilon})$ bits of space, by Theorem 4. Therefore, the total space complexity of Algorithm 4 is $O(\frac{\log^2 n}{\epsilon^7\delta^3} \log \frac{1}{\epsilon\delta})$ bits.

Next, we show the approximation guarantee of Algorithm 4. Similar to arbitrary order streams, we cannot detect H by the end of stream. Instead, we have $\tilde{H}$. Let $A_1 := e(\tilde{L}^+, \tilde{L}^-) + \sum_{v \in \tilde{H}} \max\{e(v, \tilde{L}^-), e(v, \tilde{L}^+)\}$ denote the value of the cut $(\tilde{C}, V \setminus \tilde{C})$ obtained by running Algorithm 1 on $G[\tilde{L}]$ and then assigning the vertices in $\tilde{H}$ in a greedy manner. Let $A_2 := \sum_{v \in \tilde{H}} (e(v, \tilde{L}^-) + e(v, \tilde{L}^+))$ denote the value of the cut $(\tilde{H}, \tilde{L})$. Suppose that the best of $(\tilde{C}, V \setminus \tilde{C})$ and $(\tilde{H}, \tilde{L})$ cuts is a $(\frac{1}{2} + \frac{\epsilon^2}{16})$-approximation for the MAX-CUT value of G. In the following, we show that ALG_1 and ALG_2 are good approximations of A_1 and A_2, respectively. Therefore, $\max\{\mathrm{ALG}_1, \mathrm{ALG}_2\}$ returned by Algorithm 4 is also a strictly better than $\frac{1}{2}$-approximation for the MAX-CUT value of G.

Since $V^+ = \tilde{H}^+ \cup \tilde{L}^+$ and $V^- = \tilde{H}^- \cup \tilde{L}^-$, we have $e(v, V^+) = e(v, \tilde{H}^+) + e(v, \tilde{L}^+)$ and $e(v, V^-) = e(v, \tilde{H}^-) + e(v, \tilde{L}^-)$ for any vertex $v \in V$. By Lemma 17, with probability at least $1 - \frac{\delta}{4}$, for all vertices $v \in \tilde{H}$, we have

$$\begin{aligned}e(v, \tilde{H}^+) + e(v, \tilde{L}^+) \le f_v^+ \le e(v, \tilde{H}^+) + e(v, \tilde{L}^+) + 2\epsilon^7\delta^3 m, \\ e(v, \tilde{H}^-) + e(v, \tilde{L}^-) \le f_v^- \le e(v, \tilde{H}^-) + e(v, \tilde{L}^-) + 2\epsilon^7\delta^3 m.\end{aligned}$$

Since $|\tilde{H}| \leq \frac{2\beta}{\delta^3\epsilon^4}$, we have $0 \leq e(v,\tilde{H}^+), e(v,\tilde{H}^-) \leq \frac{2\beta}{\delta^3\epsilon^4}$. Therefore, we have

$$e(v,\tilde{L}^+) \leq f_v^+ \leq \frac{2\beta}{\delta^3\epsilon^4} + 2\epsilon^7\delta^3 m + e(v,\tilde{L}^+),$$
$$e(v,\tilde{L}^-) \leq f_v^- \leq \frac{2\beta}{\delta^3\epsilon^4} + 2\epsilon^7\delta^3 m + e(v,\tilde{L}^-).$$

Then we have

$$0 \leq f_v^+ - e(v,\tilde{L}^+) \leq \frac{2\beta}{\delta^3\epsilon^4} + 2\epsilon^7\delta^3 m \quad \text{and} \quad 0 \leq f_v^- - e(v,\tilde{L}^-) \leq \frac{2\beta}{\delta^3\epsilon^4} + 2\epsilon^7\delta^3 m,$$

and

$$\max\{e(v,\tilde{L}^+), e(v,\tilde{L}^-)\} \leq \max\{f_v^+, f_v^-\} \leq \frac{2\beta}{\delta^3\epsilon^4} + 2\epsilon^7\delta^3 m + \max\{e(v,\tilde{L}^+), e(v,\tilde{L}^-)\}.$$

Note that

$$e(\tilde{L}^+,\tilde{L}^-) = e(V^+,V^-) - e(\tilde{H}^+,\tilde{H}^-) - \sum_{v\in\tilde{H}^+} e(v,\tilde{L}^-) - \sum_{v\in\tilde{H}^-} e(v,\tilde{L}^+),$$
$$\tilde{e}(\tilde{L}^+,\tilde{L}^-) = e(V^+,V^-) - \sum_{v\in\tilde{H}^+} f_v^- - \sum_{v\in\tilde{H}^-} f_v^+.$$

We have

$$\begin{aligned} e(\tilde{L}^+,\tilde{L}^-) - \tilde{e}(\tilde{L}^+,\tilde{L}^-) &= \sum_{v\in\tilde{H}^+}(f_v^- - e(v,\tilde{L}^-)) + \sum_{v\in\tilde{H}^-}(f_v^+ - e(v,\tilde{L}^+)) - e(\tilde{H}^+,\tilde{H}^-) \\ &\leq \frac{2\beta}{\delta^3\epsilon^4}\cdot\left(\frac{2\beta}{\delta^3\epsilon^4} + 2\epsilon^7\delta^3 m\right) + \frac{2\beta}{\delta^3\epsilon^4}\cdot\left(\frac{2\beta}{\delta^3\epsilon^4} + 2\epsilon^7\delta^3 m\right) - 0 \\ &= \frac{8\beta^2}{\delta^6\epsilon^8} + 8\beta\epsilon^3 m. \end{aligned}$$

Therefore,

$$\begin{aligned} A_1 - \mathrm{ALG}_1 &= (e(\tilde{L}^+,\tilde{L}^-) - \tilde{e}(\tilde{L}^+,\tilde{L}^-)) + \sum_{v\in\tilde{H}}(\max\{e(v,\tilde{L}^-), e(v,\tilde{L}^+)\} - \max\{f_v^-, f_v^+\}) \\ &\leq \left(\frac{8\beta^2}{\delta^6\epsilon^8} + 8\beta\epsilon^3 m\right) + \frac{2\beta}{\delta^3\epsilon^4}\cdot 0 = \frac{8\beta^2}{\delta^6\epsilon^8} + 8\beta\epsilon^3 m. \end{aligned}$$

So, $\mathrm{ALG}_1 \geq A_1 - (\frac{8\beta^2}{\delta^6\epsilon^8} + 8\beta\epsilon^3 m) = A_1 - \Theta(\epsilon^3 m)$.

Similarly,

$$A_2 - \mathrm{ALG}_2 = \sum_{v\in\tilde{H}}((e(v,\tilde{L}^-) - f_v^-) + (e(v,\tilde{L}^+) - f_v^+)) \leq \frac{2\beta}{\delta^3\epsilon^4}\cdot(0+0) = 0.$$

So, $\mathrm{ALG}_2 \geq A_2$.

Since we assume that $\max\{A_1, A_2\}$ is a $(\frac{1}{2} + \frac{\epsilon^2}{16})$-approximation for the MAX-CUT value of G, we have $\mathrm{ALG} := \max\{\mathrm{ALG}_1, \mathrm{ALG}_2\} \geq \max\{A_1, A_2\} - \Theta(\epsilon^3 m) \geq (\frac{1}{2} + \frac{\epsilon^2}{16})\cdot\mathrm{OPT} - \Theta(\epsilon^3\cdot\mathrm{OPT}) = (\frac{1}{2} + \Omega(\epsilon^2))\cdot\mathrm{OPT}$.

Finally, it remains to show that the best of $(\tilde{C}, V\setminus\tilde{C})$ and $(\tilde{H},\tilde{L})$ cuts is a $(\frac{1}{2} + \frac{\epsilon^2}{16})$-approximation for the MAX-CUT value of G. This directly follows from Theorem 6 (with failure probability $\frac{\delta}{4}$), by substituting H and L with $\tilde{H}$ and $\tilde{L}$, respectively. Together with Lemma 15, Lemma 16, Lemma 17, and applying union bound, this concludes the proof. ◀

B Constant Query Complexity in Random Order Streams

In this section, we show that in the case of random-order streams, it suffices to query the ϵ-accurate oracle $\mathcal{O}$ for the labels of a constant number of vertices. For simplicity, we provide only a sketch of the main ideas and omit the formal proof. We will utilize the following lemma.

▶ **Lemma 18** (Lemma 2.2 in [9]). *Let $a, b \in \mathbb{R}$ with $a < b$, $n \in \mathbb{N}$ and $x \in [a,b]^n$. Let $\Sigma = \sum_{i=1}^{n} x_i$. For any $\eta, \delta \in (0,1)$, there exists an algorithm which samples a set T of $t = O(\eta^{-2} \log \delta^{-1})$ indices from $[n]$ and returns an estimate $\tilde{\Sigma} = \frac{n}{t} \sum_{i \in T} x_i$ such that $|\Sigma - \tilde{\Sigma}| \leq \eta(b-a)n$ with probability at least $1 - \delta$.*

Similar to Algorithm 2, we store the first $\text{poly}(1/\epsilon, 1/\delta)$ edges in the random order stream and use them to identify a set $\tilde{H}$ of size $\text{poly}(1/\epsilon, 1/\delta)$ that contains all high-degree vertices H with high probability as in Lemma 10. For this part, we do not need any information on the labels of vertices provided by the oracle $\mathcal{O}$. Recall that the algorithm considers two candidate cuts and returns the one with the larger size. Let $\tilde{L} := V \setminus \tilde{H}$ and $\tilde{S} := \tilde{H} \setminus H$. The first candidate is obtained by performing the greedy extension of $(\tilde{L}^+ \cup \tilde{S}^+, \tilde{L}^- \cup \tilde{S}^-)$ using H. The second candidate is simply the cut (H, L). Formally, the sizes of these two cuts are given as follows:

$$\begin{aligned}
\text{ALG}_1 &= e(L^+, L^-) + \sum_{v \in H} \max\{e(v, L^-), e(v, L^+)\} \\
&= e(\tilde{L}^+, \tilde{L}^-) + e(\tilde{S}^+, \tilde{S}^-) + \sum_{v \in \tilde{S}^+} e(v, \tilde{L}^-) + \sum_{v \in \tilde{S}^-} e(v, \tilde{L}^+) + \sum_{v \in H} \max\{e(v, L^-), e(v, L^+)\}, \\
\text{ALG}_2 &= e(H, L) = \sum_{v \in H} e(v, L) = \sum_{v \in H} (e(v, \tilde{L}) + e(v, \tilde{S})).
\end{aligned}$$

Observe that, to compute the second cut size ALG_2, there is no need to query the oracle $\mathcal{O}$. It suffices to count $e(v, \tilde{L})$ for each vertex $v \in \tilde{H}$ using the remaining edges (after the first $\text{poly}(1/\epsilon, 1/\delta)$ edges) during the stream. At the end of the stream, we can retrieve H from $\tilde{H}$ and compute ALG_2 exactly.

Next, we focus on estimating ALG_1 by querying the oracle $\mathcal{O}$ a constant number of times. Specifically, we decompose ALG_1 into four parts and estimate each part respectively.

- **(I)** $e(\tilde{L}^+, \tilde{L}^-)$. During the stream (after the first $\text{poly}(1/\epsilon, 1/\delta)$ edges), we employ reservoir sampling to sample $O(\epsilon^{-4} \log \delta^{-1})$ edges T uniformly at random from $E(G[\tilde{L}])$, the set of edges with both endpoints in $\tilde{L}$. Let $U := \bigcup_{(u,v) \in T} \{u, v\}$. By Lemma 18, we query $\mathcal{O}$ for the vertices in U and use $\frac{e(G[\tilde{L}])}{|T|} \cdot e(U^+, U^-)$ to estimate $e(\tilde{L}^+, \tilde{L}^-)$, with an additive error of $\epsilon^2 \cdot e(G[\tilde{L}])$.
- **(II)** $e(\tilde{S}^+, \tilde{S}^-)$. Since we store all edges with both endpoints in $\tilde{H}$ during the stream and $|\tilde{H}| = \text{poly}(1/\epsilon, 1/\delta)$, we can compute $e(\tilde{S}^+, \tilde{S}^-)$ exactly by querying $\mathcal{O}$ a constant number of times.
- **(III)** $\sum_{v \in \tilde{S}^+} e(v, \tilde{L}^-) + \sum_{v \in \tilde{S}^-} e(v, \tilde{L}^+)$. For each vertex $v \in \tilde{H}$, we use reservoir sampling to sample $O(\epsilon^{-4} \log \delta^{-1})$ edges T_v uniformly at random from $E(v, \tilde{L})$, the set of edges with one endpoint at v and the other in $\tilde{L}$. Let $U_v := \bigcup_{(u,v) \in T_v} \{u\}$. By Lemma 18, we query $\mathcal{O}$ for the vertices in U_v and use $\frac{e(v, \tilde{L})}{|T_v|} \cdot e(v, U_v^+)$ to estimate $e(v, \tilde{L}^+)$ and use $\frac{e(v, \tilde{L})}{|T_v|} \cdot e(v, U_v^-)$ to estimate $e(v, \tilde{L}^-)$, both with an additive error of $\epsilon^2 \cdot e(v, \tilde{L})$. Therefore, we can estimate $\sum_{v \in \tilde{S}^+} e(v, \tilde{L}^-) + \sum_{v \in \tilde{S}^-} e(v, \tilde{L}^+)$ with an additive error of $\epsilon^2 \cdot \sum_{v \in \tilde{S}} e(v, \tilde{L}) = \epsilon^2 \cdot e(\tilde{S}, \tilde{L})$.

(IV) $\sum_{v\in H}\max\{e(v,L^-),e(v,L^+)\}$. Since $L^- = \tilde{S}^- \cup \tilde{L}^-$ and $L^+ = \tilde{S}^+ \cup \tilde{L}^+$, we have $e(v,L^-) = e(v,\tilde{S}^-) + e(v,\tilde{L}^-)$ and $e(v,L^+) = e(v,\tilde{S}^+) + e(v,\tilde{L}^+)$, for each vertex $v \in H$. Similar to **(II)** and **(III)**, we can estimate $\sum_{v\in H}\max\{e(v,L^-),e(v,L^+)\}$ with an additive error of $\epsilon^2 \cdot \sum_{v\in H} e(v,\tilde{L}) = \epsilon^2 \cdot e(H,\tilde{L})$.

Let $\widetilde{\mathrm{ALG}}_1$ denote our estimator for ALG_1. Then we have $|\widetilde{\mathrm{ALG}}_1 - \mathrm{ALG}_1| \le \epsilon^2 \cdot (e(G[\tilde{L}]) + e(\tilde{S},\tilde{L}) + e(H,\tilde{L})) \le \epsilon^2 \cdot (e(G[L]) + e(H,L)) \le \epsilon^2 m$.

If $\mathrm{ALG}_2 = e(H,L) \ge (\frac{1}{2}+\epsilon^2)\cdot \mathrm{OPT}$, then we are done. Hence, without loss of generality, we assume that $\mathrm{ALG}_2 = e(H,L) < (\frac{1}{2}+\epsilon^2)\cdot \mathrm{OPT}$. By the proof of Theorem 6, we have $\mathrm{ALG}_1 \ge (\frac{1}{2}+\frac{\epsilon^2}{16})\cdot \mathrm{OPT} \ge (\frac{1}{2}+\frac{\epsilon^2}{16})\cdot \frac{m}{2} \ge \epsilon m$ if we set $\epsilon \le \frac{1}{4}$. Therefore, $|\widetilde{\mathrm{ALG}}_1 - \mathrm{ALG}_1| \le \epsilon^2 m \le \epsilon \cdot \mathrm{ALG}_1$. Since $\mathrm{ALG}_1 \ge (\frac{1}{2}+\frac{\epsilon^2}{16})\cdot \mathrm{OPT}$ and we can use $\widetilde{\mathrm{ALG}}_1$ to approximate ALG_1 within a multiplicative factor of $(1\pm\epsilon)$, we are done.

Query Complexity. In the above analysis, the total query complexity for the ϵ-accurate oracle $\mathcal{O}$ is $\mathrm{poly}(1/\epsilon, 1/\delta, \log(1/\delta))$, which is independent of n.

Distributed and Parallel Low-Diameter Decompositions for Arbitrary and Restricted Graphs

Jinfeng Dou
Paderborn University, Germany

Thorsten Götte
University of Hamburg, Germany

Henning Hillebrandt
Paderborn University, Germany

Christian Scheideler
Paderborn University, Germany

Julian Werthmann
Paderborn University, Germany

Abstract

We consider the distributed and parallel construction of low-diameter decompositions with strong diameter. We present algorithms for arbitrary undirected, weighted graphs and also for undirected, weighted graphs that can be separated through $k \in \tilde{O}(1)$ shortest paths. This class of graphs includes planar graphs, graphs of bounded treewidth, and graphs that exclude a fixed minor K_r. Our algorithms work in the PRAM, CONGEST, and the novel HYBRID communication model and are competitive in all relevant parameters. Given $\mathcal{D} > 0$, our low-diameter decomposition algorithm divides the graph into connected clusters of strong diameter $\mathcal{D}$. For an arbitrary graph, an edge $e \in E$ of length ℓ_e is cut between two clusters with probability $O(\frac{\ell_e \cdot \log(n)}{\mathcal{D}})$. If the graph can be separated by $k \in \tilde{O}(1)$ paths, the probability improves to $O(\frac{\ell_e \cdot \log\log n}{\mathcal{D}})$. In either case, the decompositions can be computed in $\tilde{O}(1)$[1] depth and $\tilde{O}(m)$ work in the PRAM and $\tilde{O}(1)$ time in the HYBRID model. In CONGEST, the runtimes are $\tilde{O}(\mathrm{HD} + \sqrt{n})$ and $\tilde{O}(\mathrm{HD})$ respectively. All these results hold w.h.p.[2]

Broadly speaking, we present distributed and parallel implementations of sequential divide-and-conquer algorithms where we replace exact shortest paths with approximate shortest paths. In contrast to exact paths, these can be efficiently computed in the distributed and parallel setting [STOC '22]. Further, and perhaps more importantly, we show that instead of explicitly computing vertex-separators to enable efficient parallelization of these algorithms, it suffices to sample a few random paths of bounded length and the nodes close to them. Thereby, we do not require complex embeddings whose implementation is unknown in the distributed and parallel setting.

2012 ACM Subject Classification Theory of computation → Distributed algorithms; Theory of computation → Parallel algorithms; Mathematics of computing → Graph algorithms

Keywords and phrases Distributed Graph Algorithms, Network Decomposition, Excluded Minor

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.45

Related Version *Full Version*: `https://arxiv.org/abs/2411.19859` [24]

Funding J. Dou acknowledges the support by the China Scholarship Council under grant number 202006220268. T. Götte acknowledges the support by DFG grant 491453517.

[1] $\tilde{O}(\cdot)$ hides polylogarithmic factors in n.
[2] We say an event holds w.h.p. if it occurs with probability $1 - o(n^c)$ for some $c > 1$.

© Jinfeng Dou, Thorsten Götte, Henning Hillebrandt, Christian Scheideler, and Julian Werthmann;
licensed under Creative Commons License CC-BY 4.0

1 Introduction

This paper considers the efficient construction of so-called (probabilistic) low-diameter decompositions (LDD) for general and restricted graphs in the CONGEST, PRAM, and HYBRID model. An LDD of a (weighted) graph $G := (V, E, \ell)$ with strong diameter $\mathcal{D}$ is a partition of G into disjoint connected subgraphs $\mathcal{K}(G) := K_1, \dots, K_N$ with $K_i := (V_i, E_i)$. We will refer to these subgraphs as *clusters*. We say that a cluster K_i has *strong* diameter $\mathcal{D}$ if it has a diameter of at most $\mathcal{D}$. That is, between two nodes in K_i there is a path of length (at most) $\mathcal{D}$ that only consists of nodes in K_i. In contrast, an LDD with a weak diameter creates possibly disconnected subgraphs with a diameter of $\mathcal{D}$ with respect to the original graph G. Here, for all nodes $v, w \in K_i$, there is a path of length (at most) $\mathcal{D}$ from v to w that can use nodes outside of K_i. We measure the decomposition quality by the number of edges that are *cut* between clusters. An edge $\{v, w\} \in E$ is cut if its two endpoints v and w are assigned to different clusters. Thus, a good clustering algorithm only ensures that *most* nodes are in the same cluster as their neighbors. Without this constraint, a trivial algorithm could simply remove all edges from the graph and return clusters containing one node each. There are several ways to count the edges that are cut. In this work, we consider so-called *probabilistic* decompositions. Here, the probability for an edge $z = \{v, w\}$ to be cut between two clusters K_i and K_j with $i \neq j$ that contain its respective endpoint depends on its length ℓ_z. This means that *short* edges are cut less likely than *long* edges. We use the following formal definition of LDD's in the remainder of this work:

▶ **Definition 1** (Probabilistic Low-diameter Decomposition (LDD)). *Let $G := (V, E, \ell)$ be a weighted graph and $\mathcal{D} > 0$ be a distance parameter. Then, an algorithm for computing an LDD with quality $\alpha \geq 1$ creates a series of disjoint clusters $\mathcal{K} := K_1, K_2, \dots, K_N$ with $K_i := (V_i, E_i)$ where $V_1 \sqcup \dots \sqcup V_N = V$. Futher, it holds:*

1. *Each cluster $K_i \in \mathcal{K}$ has a strong diameter of at most $\mathcal{D}$.*
2. *Each edge $z := \{v, w\} \in E$ of length ℓ_z is cut between clusters with probability $O\left(\frac{\ell_z \cdot \alpha}{\mathcal{D}}\right)$.*

If we do not require each node to be in a cluster, we use the related notion of a Low-Diameter *Clustering* (LDC). In a LDC with quality (α, β) and strong diameter $\mathcal{D}$ we create a series of disjoint clusters $\mathcal{K} := K_1, K_2, \dots, K_N$ with $K_i := (V_i, E_i)$. Each cluster $K_i \in \mathcal{K}$ has a strong diameter of at most $\mathcal{D}$ and each edge $z := \{v, w\} \in E$ of length ℓ_z is cut between clusters with probability (at most) $O\left(\frac{\ell_z \cdot \alpha}{\mathcal{D}}\right)$. Further, each node is part of a cluster with probability at least β. As we will see, LDD's and LDC's are nearly equivalent as we can build an LDD from an LDC by repeatedly applying it until all nodes are clustered.

It is known that the best possible value of α is $O(\log n)$ for general graphs. However, if the graph class is restricted, better decompositions are possible. So, alongside solutions for general graphs, this work will consider the restricted class of so-called k-path separable graphs that Abraham and Gavoille introduced in [2]. Roughly speaking, a weighted graph $G = (G, E, \ell)$ is k-path separable if the *recursive* removal of k shortest paths results in connected components containing a constant fraction of the nodes. Moreover, a graph $G = (G, E)$ is *universally* k-path separable if it is k-path separable for *every* weight function ℓ. Note that this makes universal k-path separability a topological property rather than one that depends on the weight function. Abraham and Gavoille and Diot and Gavoille formalized this as follows:

▶ **Definition 2** (Universally k-Path Separarable Graphs [2, 22]). *A weighted graph $G := (V, E, \ell)$ with n vertices is* k-path separable*, if there exists a subset of vertices S, called a k-path separator, such that:*

1. $S := \mathcal{P}_0 \cup \mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots$, *where each* $\mathcal{P}_i := \{P_{i_1}, P_{i_2}, \ldots\}$ *is a set of shortest paths in* $G \setminus \bigcup_{j<i} \mathcal{P}_j$.
2. *The total number of paths in* S *is at most* k, *i.e.,* $\sum_{\mathcal{P}_i \in S} |\mathcal{P}_i| \leq k$.
3. *Each connected component in* $G \setminus S$ *contains at most* $\frac{n}{2}$ *nodes.*
4. *Either* $G \setminus S$ *is empty or* k*-path separable.*

If G *is* k*-path separable for any weight function, we call* G *universally* k*-path separable. We will sometimes abuse notation and use* $P_{i_j} \in S$ *when we refer to some path* $P_{i_j} \in \mathcal{P}_i$ *and* $\mathcal{P}_i \in S$.

In this work, we will only consider universally k-path separable graphs. Given this definition, a natural question is which graph classes are k-path separable. Clearly, all graphs that have separators that consist of at most η vertices are trivially universally η-path separable. This follows because every node can be seen as the shortest path to itself in any subgraph that it is contained in. Notably, graphs of bounded tree-width τ have separators that consist of τ nodes [53] and are therefore universally τ-path separable. Further, due to Thorup, it is known that planar graphs are universally 3-path separable [58]. More generally, Abraham and Gavoille [2] showed that every graph $G := (V, E, \ell)$ that does not include a fixed clique minor K_r is universally k-path separable where $k := f(r)$ depends only on r and not the size of G. The concrete value of $f(r)$ is based on the Structure Theorem by Robertson and Seymour [54].

These restricted graph classes have received significant attention in the context of distributed algorithms (cf. [35, 36, 37, 39, 42]). Not only do they contain graph topologies that frequently appear in practice, they also have favorable properties for distributed algorithms as they exclude pathologic worst case topologies. As k-path separable graphs are a superclass of these restricted graphs, we derive novel results for planar graphs, graphs of bounded treewidth, and most importantly, graphs that exclude a fixed minor K_r.

In a nutshell, prior works [13] and [55] have shown that computing LDD's for general graphs can be broken down into $\tilde{O}(1)$ applications of your favorite approximate shortest path algorithm in your favorite model of computation. We extend this innovative insight in two ways by improving the algorithm of [13] and showing that a) we can construct a strong diameter LDD of quality $O(\log n)$ for general graphs and b) on k-path separable graphs one can compute LDD's of quality $O(\log k + \log\log n)$. Again, our algorithms use only approximate shortest paths and are therefore applicable to various models.

Due to its many technical results, this paper takes the form of an extended abstract. In the main part of the paper, we present our algorithms, state our main lemmas, and sketch their proofs. More detailed descriptions and the full analysis, including all proofs, can be found in the full version on arXiv [24]. In the remainder of this section, we will first introduce some notations and convention in Subsection 1.1, then introduce our generalized model of computation in Subsection 1.2, and and state our main results in Subsection 1.3. Finally, we present some important applications for LDD's in Subsection 1.4 and discuss further related work (with a special focus on CONGEST algorithms) in Subsection 1.5.

1.1 Preliminaries and Notations

In this work, we only consider *undirected* graphs $G = (V, E)$ unless specifically mentioned otherwise. We further assume some basic familiarity with graph theory. For an introduction, we refer to standard textbooks, e.g., [21]. Throughout this work, we denote the distance between a node $v \in V$ and a set $S \subset V$ as $d(v, S)$. This distance is defined as the (weighted) length of the shortest path between v and the node $w \in S$ closest to v. If we consider the distance w.r.t. to a subgraph $H \subset G$, we write $d_H(\cdot, \cdot)$. Further for a subset $S \subseteq V$, a

subgraph $H \subseteq G$, and a distance δ, we define $B_H(S, \delta) = \{v \in V \mid d_H(v, S) \leq \delta\}$ to be the so-called *ball* that contains all nodes in distance (at most) δ to any node in G. Again, for $H = G$, we omit the subscript.

1.2 Model(s)

In this work, we present algorithms that can be implemented in several different models of computation for parallel and distributed systems by reducing our algorithms to (approximate) shortest-path computations. Notably, we can derive algorithms for the CONGEST and the PRAM models, which are the de facto standard models for distributed and parallel computing, respectively.

PRAM: In the PRAM model, we assume computations are executed on a machine with p processors. The processes work over a set of shared memory cells M. The input graph G is stored in these cells. In a single step of computation, they can read and write from each cell in an atomic operation. If more than one processor writes in a cell, an arbitrary processor succeeds, i.e., we consider a CRCW PRAM. The *work* of a PRAM algorithm is the total number of primitive read or write operations that the processors perform. Further, the *depth* is the length of the longest series of operations that have to be performed sequentially.

CONGEST: In the CONGEST model [52], we consider a static graph $G = (V, E)$. Each node has a unique identifier. We assume these identifiers consist of $O(\log n)$ bits, i.e., they are picked from interval $[1, \dots, n^c]$ for some $c \in \Theta(1)$. Time proceeds in synchronous rounds. In each round, nodes receive *all* messages sent in the previous round; they perform (unlimited) computation based on their internal states and the messages they have received so far. Finally, a node $v \in V$ may send a *distinct* message of size $O(\log n)$ to each neighbor in G. Thus, a message may contain a node's identifier and a few more bits of information.

HYBRID: In addition to these two *classic* models of computation, we will also use the recently established HYBRID model. The HYBRID model was introduced in [7] as a means to study distributed systems that leverage multiple communication modes. Just as in CONGEST, we assume synchronous rounds. In the HYBRID model, a node can send a message to all its neighbors in G in the so-called *local* mode and also $O(\log n)$ nodes of its choice (as long as no node receives more than $O(\log n)$ messages) in the so-called *global* mode. One typically parameterizes the size of the messages in the local mode. For a discussion of this model, we refer the interested reader to [7, 57].

Rather than exploiting these models' intricacies in bespoke algorithms, we reduce our algorithms to $\tilde{O}(1)$ applications of a $(1+\epsilon)$-approximate shortest path algorithm with $\epsilon \in O\left(1/\log^2 n\right)$ and some simple aggregations on subsets of the input graph. To be precise, our algorithms are based on approximate set-source shortest paths (SetSSP) and so-called minor aggregations. In the remainder of this section, we will clarify what exactly these operations do and how efficiently they can be implemented in the CONGEST, the PRAM, and the HYBRID model.

First, we formalize the class of approximate SetSSP algorithms. For our algorithms, we must be able to compute the shortest paths to a subset of nodes and also to virtual nodes that may not be part of the original input graphs. We define this problem as follows:

▶ **Definition 3** (Approximate SetSSP with Virtual Nodes). *Let $G := (V, E)$ be a weighted graph and let $G' := (V', E', w)$ with $V' := V \cup \{s_1, s_2, \dots\}$ be the graph that results from adding $\tilde{O}(1)$ virtual nodes to G. Each virtual node can have an edge of polynomially bounded weight to every virtual and non-virtual node in G'. Finally, let $S \subset V'$ be an arbitrary subset of virtual and non-virtual nodes. Then, an algorithm that solves $(1+\epsilon)$-approximate set-source shortest path with virtual nodes computes the following:*

- *Each node $v \in V' \setminus S$ learns a predecessor $p_v \in N(v)$ on a path of length at most $(1+\epsilon)d(v,S)$ to some node in S and marks the edge $\{v, p_v\}$. Together, all the marked edges imply an approximate shortest path tree T[3] rooted in set S.*
- *Each node $v \in V'$ learns its distance $d_T(v,S) \leq (1+\epsilon)d(v,S)$ to S in tree T, i.e., its exact distance to S in T and its $(1+\epsilon)$-approximate distance to S in G'.*

Note that, if we can compute SetSSP on a graph G in τ time/depth, we can also compute it on any set of disjoint subgraphs of G in τ time/depth. To be precise, let $C_1, \dots, C_N$ with $C_i = (V_i, E_i)$ be a set of disjoint subgraphs of G and let $S_1, \dots, S_N$ with $S_i \subset V_i$ be subsets of nodes from each C_i. Suppose, we want to compute an SetSSP from each S_i in C_i. To this end, we simply assign a prohibitively high length to an edge between two subgraphs $C_i \neq C_j$. Then, we compute a SetSSP from $\mathcal{S} = \{S_1, \dots, S_N\}$ that results in a tree T. Due to their length, no (approximate) shortest path will ever contain an edge between subgraphs and so, the closest node to any node in C_i must be from S_i. Thus, the tree restricted to C_i is an approximate shortest path tree for S_i.

Our second building block are so-called *minor aggregations*, which were first introduced in [35, 36]. Consider a network $G = (V, E)$ and a (possibly adversarial) partition of vertices into disjoint subsets $V_1, V_2, \dots, V_N \subset V$, each of which induces a *connected* subgraph $G[V_i]$. We will call these subsets *minors*. Further, let each node $v \in V$ have private input x_v of length $\tilde{O}(1)$, i.e., a value that can be sent along an edge in $\tilde{O}(1)$ rounds. Finally, we are given an aggregation function $\bigotimes$ like `SUM`, `MIN`, `AVG`, ..., which we want to compute in each part. Then a *minor aggregation* computes these functions in all minors $G[V_i]$. Note that the diameter of these minors might be (much) larger than the diameter of G. Therefore, the corresponding algorithm has to use edges outside of each minor to be efficient.

Both approximate shortest paths and minor aggregation can be efficiently implemented in arbitrary and restricted graphs in all of our models. Note that we are mainly interested in algorithms that require $\tilde{O}(1)$ aggregations and $(1+\epsilon)$-approximate SetSSP computations with $\epsilon \in \Omega(1/\log^c n)$ as these algorithms can then be efficiently implemented in all three models. Note that *efficient* means different things in different models. While in the PRAM, we hope for a polylogarithmic depth using a linear number of processors and in HYBRID we aim for polylogarithmic runtime, we have to manage our expectations for CONGEST. In the CONGEST model, the best we can hope for is $O(\mathrm{HD})$, where HD denotes the length of the longest shortest path between two nodes *if we ignore the weights*, the so-called hop diameter. The hop-diameter is a natural lower bound for most *global* problems in CONGEST as it describes the precise time in which a message starting from a node $v \in V$ can (theoretically) reach every node in the network. While this can be up to n, it may be much smaller in many practical graphs.

To be precise, it holds:

▶ **Theorem 4.** *Let $G := (V, E, \ell)$ be a (weighted) graph and let $\mathcal{A}$ be an algorithm that can be broken down into the following two types of operations:*

1. *A $(1+\epsilon)$-approximate SetSSP on a set of disjoint connected subgraphs $C_1, \dots, C_N$ with $C_i = (V_i, E_i)$ from a set of sources $S_1, \dots, S_N$ with $S_i \subset V_i$.*
2. *A minor aggregation on a set of disjoint connected subgraphs $C_1, \dots, C_N$ with $C_i = (V_i, E_i)$.*

[3] Technically, for a set S with $|S| \geq 2$, the algorithm produces a forest with the individual tree rooted in the nodes S. However, we slightly abuse notation and refer to it as a forest.

The subgraphs $C_1, \dots, C_N$ may differ in each step. Suppose that $\mathcal{A}$ can be broken down into τ_s steps of $(1+\epsilon)$-approximate SetSSP computations and τ_m minor aggregations. Then, it holds:

- *In* CONGEST*, $\mathcal{A}$ can be executed in $\tilde{O}\left((\epsilon^{-2} \cdot \tau_s + \tau_m) \cdot (HD + \sqrt{n})\right)$ time, w.h.p. If graph G is (universally) k-path separable, $\mathcal{A}$ can be executed in $\tilde{O}\left((\epsilon^{-2} \cdot \tau_s + \tau_m) \cdot k \cdot HD\right)$ time, w.h.p.*
- *In the* PRAM *model with $\tilde{O}(m)$ processors, $\mathcal{A}$ can be executed with $\tilde{O}\left((\epsilon^{-2} \cdot \tau_s + \tau_m) \cdot m\right)$ work and $\tilde{O}\left(\tau_s + \tau_m\right)$ depth, w.h.p.*
- *In the* HYBRID *model, $\mathcal{A}$ can be executed in $\tilde{O}\left((\epsilon^{-2} \cdot \tau_s + \tau_m)\right)$ time, w.h.p., using $O(n)$ bits per message in the local mode. If graph G is (universally) k-path separable, $\mathcal{A}$ can be executed in $\tilde{O}\left((\epsilon^{-2} \cdot \tau_s + \tau_m) \cdot k\right)$ time, w.h.p., using $O(\log n)$ bits per message in the local mode*

For general graphs, the bounds for approximate SetSSP in CONGEST and PRAM follow from a recent breakthrough result [56]. Similarly, the runtimes for minor aggregation in CONGEST and PRAM are given in [41] and [56], respectively. Finally, [57] showed the bounds for both SetSSP and minor aggregation in the HYBRID model. For the runtime on k-path separable graphs in CONGEST, we use that k-path separable graphs exclude K_{4k+1} as minor [22]. If the graph excludes a fixed clique minor K_r, the runtime of a minor aggregation is $\tilde{O}(r \cdot \mathrm{HD})$ [37] and $(1+\epsilon)$-approximate SetSSP can be solved in $\tilde{O}(\epsilon^{-2} \cdot r \cdot \mathrm{HD})$ [56]. Note that for CONGEST, all these bounds are optimal as neither SetSSP nor the aggregation can be computed faster [41].

1.3 Our Contributions

In the following, we give a formal statement of our contributions. First, we show that we can achieve the (asymptotically) best possible quality of $O(\log n)$ for general through approximate SetSSP and minor aggregation. Our first main theorem is the following.

▶ **Theorem 5** (LDDs for General Graphs)**.** *Let $\mathcal{D} > 0$ be an arbitrary distance parameter and $G := (V, E, \ell)$ be a (possibly weighted) undirected graph. Then, there is an algorithm that creates an LDD of G with strong diameter $\mathcal{D}$ and quality $O(\log n)$. The algorithm can be implemented with $\tilde{O}(1)$ minor aggregations and $\tilde{O}(1)$ $(1+\epsilon)$-approximate SetSSP computations where $\epsilon \in O\left(1/\log^2 n\right)$.*

By Theorem 4, this implies that the algorithm can, w.h.p., be implemented in $\tilde{O}(\mathrm{HD} + \sqrt{n})$ time in CONGEST, and $\tilde{O}(1)$ depth in the PRAM and $\tilde{O}(1)$ time in HYBRID. Thus, our algorithm is currently the best randomized LDD construction in CONGEST for general weighted graphs. It has same runtime as [55] and [13], creates clusters of *strong* diameter, and has (asymptotically) optimal quality of $O(\log n)$. We achieve this improvement through a more fine-grained analysis of the well-known exponential-delay clustering first used by Miller et al. [50] that was already used by Becker, Lenzen, and Emek in [13]. In fact, we only present a little addition to their existing algorithm to ensure the strong diameter.

Second, for k-path separable graphs with $k \in \tilde{O}(1)$, we present an algorithm with almost exponentially better quality. More precisely, our second main theorem is the following.

▶ **Theorem 6** (LDDs for k-Path Seperable Graphs)**.** *Let $\mathcal{D} > 0$ be an arbitrary distance parameter and $G := (V, E, \ell)$ be a (possibly weighted) $\tilde{O}(1)$-path separable graph. Then, there is an algorithm that creates an LDD of G with strong diameter $\mathcal{D}$ and quality $O(\log \log n)$. The algorithm can be implemented with $\tilde{O}(1)$ minor aggregations and $\tilde{O}(1)$ $(1+\epsilon)$-approximate SetSSP computations where $\epsilon \in O\left(1/\log^2 n\right)$.*

Thus, by the same reasoning as above, the algorithm can, w.h.p., be implemented in $\tilde{O}(\mathrm{HD})$ time in CONGEST, and $\tilde{O}(1)$ depth in the PRAM and $\tilde{O}(1)$ time in HYBRID.

Our main technical novelty for proving Theorem 6 is a weaker form of a path separator that can be constructed without computing an embedding of G. Prior algorithms for k-path separable graphs, e.g. [2], often employ a so-called shortest path decomposition where we recursively compute k-path separators until the graph is empty. However, computing these separators requires precomputing a complex embedding, which we cannot afford in a distributed or parallel setting. Instead, we take a different approach to completely avoid the (potentially expensive) computation of an embedding. We show that by sampling approximate shortest paths (in a certain way), we can obtain what we will call a weak $\mathcal{D}$-separator. A weak $\mathcal{D}$-separator S' only ensures that in the graph $G \setminus S$, all nodes have at most a constant fraction of nodes in distance $\mathcal{D}$. In particular, in contrast to a *classical* separator, the graph $G \setminus S'$ might still be connected. We strongly remark that, in considering weak separators, we sacrifice some of the useful properties of a traditional seperator. Nevertheless, this only results in some additional polylogarithmic factors in the runtime of our applications.

1.4 Applications of LDDs

In this section, we discuss the natural question of why we consider LDD's in the first place. Simply put, LDD's are part of an algorithm designer's toolkit for developing efficient divide-and-conquer algorithms, as they offer a generic way to decompose a given graph. As such, creating LDD's with low edge-cutting probability plays a significant role in numerous algorithmic applications. In the following, we present a comprehensive list of applications (which we do not claim to be complete):

Tree Embeddings. As a first example, LDD's can be used to embed graphs into trees, i.e., constructing so-called metric tree embeddings (MTE) and low-stretch spanning trees (LSSP). In both, we map a given graph $G := (V, E, \ell)$ into a randomly sampled tree $T = (V, E_T, \ell_T)$ in a way that preserves the path lengths of the original graph G on expectation. In an LSSP, it must hold $E_T \subset E$, i.e., we compute a subtree of G, while a MTE can use virtual edges not present in the original graph. LSSPs and MTEs have proven to be a helpful design paradigm for efficient algorithms as many computational problems are significantly easier to solve on trees than on general graphs. For example, they have been used to construct distributed and parallel solvers for certain important classes of LPs [6, 14, 20, 38], which can then be used solve MaxFlow and other complex problems. Nearly all constructions use LDD's (or variants thereof) as subroutines; see [1, 5, 12, 25] for the construction of LSSPs and [8, 9, 10, 31] for MTEs.

Light Spanners. Another interesting avenue for LDD's are so-called *light spanners* first introduced by Elkin, Neimann, and Solomon in [30]. Given a weighted graph $G = (V, E, \ell)$, a t-spanner is a subgraph $H \subset G$ that approximately preserves the distances between the nodes of the original graph by a factor t, i.e., for all pairs of node $v, w \in V$, it holds $d_H(v, w) \leq t \cdot d_G(v, w)$. A spanner's lightness L_H is the ratio between the weight of all its edges and the MST of graph G, i.e., it holds $L_H = \sum_{e \in H} \ell_e / \sum_{e \in MST(G)} \ell_e$. Thus, compared to the LSSPs and MTEs, light spanners have more freedom in choosing their edges as they are not required to be trees. The lightness is (arguably) a natural quality measure for a spanner, especially for distributed computing. Consider, for example, an efficient broadcast scheme in which a node wants to spread a message to every other node in G. Further, suppose the edges' length as the cost of using this edge to transmit a message. If we send the messages along

the edges of the light spanner H of G, the total communication cost can be bound by its lightness, and the stretch bounds the cost of the path between any two nodes. Despite this, there are almost no distributed algorithms that compute t-spanners with an upper bound on the lightness with [26] being a notable exception of stretch $O(k)$ with lightness $\tilde{O}(n^{\frac{1}{k}})$. The complicating factor in the distributed construction is that lightness is a global measure. Many previously known distributed spanner constructions like [11] or [50] are very local in their computations as they only consider each node's t-neighborhood with $t \in O(\log n)$. Therefore, the resulting spanners' lightness is unbounded despite having few edges. LDD's are connected to light spanners through an observation by Neiman and Filtser in [33]: They show that with black-box access to an LDD algorithm with quality α, one can construct an $O(\alpha)$-spanner with lightness $\tilde{O}(\alpha)$ for any weighted graph $G = (V, E, \ell)$. In addition to an algorithm that creates LDD's for geometrically increasing diameters, they only require so-called (α, β)-nets for which they already provided a distributed implementation in [26]. Thus, finding better distributed algorithms LDDs, especially for restricted graphs, drectly improves the distributed construction of light spanners.

Compact Routing Schemes. *Routing schemes* are distributed algorithms that manage the forwarding of data packets between network devices. Thus, developing efficient routing schemes is essential to enhance communication among multiple parties in distributed systems and, therefore, is well researched [15, 18, 27, 28, 42, 43, 44, 45, 46, 48]. In particular, it is known that LDD's can be used to construct routing schemes with routing paths that are not significantly longer than the shortest paths in G. An LDD with strong diameter and quality α directly enables us to construct a routing scheme where all paths only differ by a factor of $O(\alpha)$ from the true shortest paths. This factor is typically referred to as the stretch of the routing scheme. The core idea is to create LDD's with diameter $O(\alpha\mathcal{D}_i)$ for geometrically increasing distance scales $\mathcal{D}_i = 2^i$. Then, for the right choice of parameters hidden in the O-notation, for any two nodes in distance $[\mathcal{D}_i, 2\mathcal{D}_i]$ there is a cluster of diameter $O(\alpha\mathcal{D})$ containing them both with constant probability. Using techniques presented in [23], this is sufficient to efficiently compute a routing scheme with stretch $O(\alpha)$ in CONGEST, the PRAM, and the HYBRID model. However, this stretch can likely be improved using more sophisticated algorithms.

1.5 Related Work

In this section, we present some related decomposition schemes. We focus on the CONGEST model as it provides the greatest variety of decomposition schemes. We present a compact overview in Table 1.

Despite the vast number of applications for LDD's on *weighted* graphs, research on LDD's in a distributed setting has mostly focused on the unweighted case, producing many efficient algorithms in this regime. Two notable exceptions, however, explicitly consider weighted graphs and are closely related to our results: The work of Becker, Emek, and Lenzen [13] creates LDD's of quality $O(\log n)$ with weak diameter[4] for general graphs. We note that $O(\log n)$ is the best quality we can hope for in an LDD due to a result by Bartal [8]. The algorithm requires $\tilde{O}(\mathrm{HD} + \sqrt{n})$ time in the CONGEST model, which is optimal as each distributed LDD construction in a weighted graph requires $\Omega(\mathrm{HD} + \sqrt{n})$ time [40]

[4] Actually, [13] proves a stronger property of the diameter. Although the diameter is weak, the number of nodes outside of a cluster that are part of the shortest path between two nodes of a cluster is limited. For many applications, this is sufficient and just as good as a strong diameter.

as we can derive approximate shortest paths from it. Just as our algorithm, [13] consists of $\tilde{O}(1)$ $(1+\epsilon)$-approximate SetSSP computations with $\epsilon \in O\left(1/\log^2 n\right)$. Further, [55], makes two significant improvements compared to [13]. They present a decomposition with strong diameter (instead of weak), and their construction is deterministic (instead of randomized). Here, they do not bound the probability that a specific edge of length ℓ is cut, but instead – since the algorithm is deterministic – count the overall number of edges of length ℓ that are cut. We say that a deterministic decomposition has quality α, if the number of edges of length ℓ with endpoint in different clusters is bound by $m \cdot \frac{\alpha \cdot \ell}{\mathcal{D}}$. Note that $m \cdot \frac{\alpha \cdot \ell}{\mathcal{D}}$ is exactly the *expected* number of cut edges in a probabilistic LDD of quality α. Thus, with constant probability, the probabilistic version roughly cuts the same number of edges. However, without further information about the specific random choices made by the corresponding algorithm, a probabilistic LDD could cut many more edges. While this can be mitigated with standard probability amplification techniques, i.e., executing the algorithm $O(\log n)$ times and picking the iteration that cuts the fewest edges, a deterministic LDD does not have this problem in the first place. In a deterministic LDD, we have the guarantee that no matter what happens in the execution of the algorithm, less than $m \cdot \frac{\alpha \cdot \ell}{\mathcal{D}}$ edges of length ℓ are cut. This makes these decompositions very useful when we do not have the time or resources for $O(\log n)$ repetitions. A typical example is distributed algorithms for local problems, where we aim for sublogarithmic runtimes. That being said, they have a slightly worse quality of only $O(\log^3 n)$. Again, the algorithm consists of $\tilde{O}(1)$ $(1+\epsilon)$-approximate SetSSP computations with $\epsilon \in O\left(1/\log^2 n\right)$.

For restricted graphs like planar, bounded treewidth, or minor-free graphs, we are unaware of a distributed/parallel algorithm explicitly designed for weighted graphs. The research [16, 17, 29, 34, 47, 19] in the CONGEST model focused on so-called *network decompositions* for unweighted graphs that enable fast algorithms for local problems like MIS, Coloring, or Matching. In principle, these algorithms could also be applied to weighted graphs and produce LDD's. However, their runtime depends on the diameter $\mathcal{D}$ of the resulting clusters, which might be much larger than the hop diameter (or even n) if the graph is weighted. We further remark that some authors use slightly different notations when describing LDD's. While our definition above focuses on the clusters' diameter, some works emphasize the number of cut edges. That is, they define it as a decomposition that cuts $\epsilon \cdot m$ edges and creates a cluster of diameter $\mathcal{D} := O(\epsilon^{-1})$. The quality is then described by the blowup in the diameter. Notably, Chang and Su [17] show that a low-diameter decomposition with $\mathcal{D} = O(r\epsilon^{-1})$ can be computed in $\epsilon^{-O(1)} \log^{O(1)} n$ rounds with high probability and $\epsilon^{-O(1)} 2^{O(\sqrt{\log n \log\log n})}$ rounds deterministically in the CONGEST model for any K_r-free graph. Chang [16] later improved the deterministic runtime to $O(\mathcal{D} \log^* n + \mathcal{D}^5)$. In particular, if we translate this to our notion of quality, Chang [16] for unweighted graphs that present LDD's in K_r-free graphs with quality $O(r)$ in the CONGEST model. Therefore, our algorithm has worse a worse quality of $O(\log\log n)$ but is faster for large diameters $\mathcal{D}$, which arguably trades off its worse cutting probability. Also note that our techniques are entirely different from those of [16]. We elaborate more on this in the full version [24].

1.6 Structure of This Paper

The main part of the paper is structured as follows: In Section 2 we present the algorithm behind Theorem 5. In doing so, we also present several useful intermediate results we reuse later. Then, in Section 3 we present our novel technique to compute weak separators. In Section 4, we combine our insights into clustering algorithms from Section 2 and findings on the construction of separators from Section 3 to develop the algorithm behind Theorem 6.

Table 1 An overview of the related work on LDD's for various graph families in the CONGEST model.

Ref.	Quality	Diam.	Runtime	Weights	Comment
[19]	$O\left(\mathcal{D}^{1/c}\right)$	Strong	$O\left(\mathcal{D}^{O(1)}\right)$	×	Planar
[47]	$O\left(\mathcal{D}^{1/c}\right)$	Strong	$\tilde{O}\left(\mathcal{D}^{O(1)}\right)$	×	K_r-free
[16]	$O(r)$	Strong	$\tilde{O}\left(\mathcal{D}^{O(1)}\right)$	×	K_r-free
Thm. 6 + [22]	$O(\log\tau + \log\log n)$	Strong	$\tilde{O}(\tau^2\text{HD})$	✓	Treewidth τ
Thm. 6 + [58]	$O(\log\log n)$	Strong	$\tilde{O}(\text{HD})$	✓	Planar
Thm. 6 + [2]	$O(\log f(r) + \log\log n)$	Strong	$\tilde{O}(r \cdot f(r) \cdot \text{HD})$	✓	K_r-free
[29]	$O(\log n)$	Strong	$\tilde{O}(\mathcal{D})$	×	
[13]	$O(\log n)$	Weak	$\tilde{O}(\text{HD} + \sqrt{n})$	✓	
[55]	$O\left(\log^3 n\right)$	Strong	$\tilde{O}(\text{HD} + \sqrt{n})$	✓	
Thm. 5	$O(\log n)$	Strong	$\tilde{O}(\text{HD} + \sqrt{n})$	✓	

2 Strong Diameter LDDs for General Graphs

In this section, we present the algorithm behind Theorem 5. The description is divided into three sections/subroutines. First, in Subsection 2.1, we present a generic clustering algorithm. Second, in Subsection 2.2 we combine this algorithm with a technique from [13] that cuts all edges with the correct probability. However, we may not add each node to a cluster. Finally, in Subsection 2.3, we present a generic technique that recursively applies the algorithm from Subsection 2.2 to cluster all nodes while asymptotically preserving the cut probability of an edge.

2.1 Pseudo-Padded Decompositions Using Approximate Shortest Paths

We begin with a crucial technical theorem that will build the foundation of most of our results. A key component in this construction is the use of truncated exponential variables. In particular, we will consider exponentially distributed random variables truncated to the $[0, 1]$-interval. Loosely speaking, a variable is truncated by resampling it until the outcome is in the desired interval. In the following, we will always talk about variables that are *truncated to the* $[0, 1]$*-interval* when we talk about truncated variables. The density function for a truncated exponential distribution with parameter $\lambda > 1$ is defined as follows:

▶ **Definition 7** (Truncated Exponential Distribution). *We say a random variable X is truncated exponentially distributed with parameter λ if and only if its density function is $f(x) := \frac{\lambda \cdot e^{-x\lambda}}{1-e^{-\lambda}}$. We write $X \sim \mathsf{Texp}(\lambda)$. Further, if $X \sim \mathsf{Texp}(\lambda)$ and $Y := \mathcal{D} \cdot X$, we write $Y \sim \mathcal{D} \cdot \mathsf{Texp}(\lambda)$.*

The truncated exponential distribution is a useful tool for decompositions that has been extensively used in the past [4, 32, 51]. Using a truncated exponential distribution and $(1+\epsilon)$-approximate SetSSP computations, we prove a helpful auxiliary result, namely:

▶ **Theorem 8** (Pseudo-Padded Decomposition for General Graphs). *Let $\mathcal{D} > 0$ be a distance parameter, ϵ be an error parameter, $G := (V, E, \ell)$ a (possibly weighted) undirected graph, and $\mathcal{X} \subseteq V$ be a set of possible cluster centers. Suppose that for each node $v \in V$, the following two properties hold:*

- ***Covering Property:*** *There is at least one $x \in \mathcal{X}$ with $d_G(v, x) \leq \mathcal{D}$.*
- ***Packing Property:*** *There are at most τ centers $x' \in \mathcal{X}$ with $d_G(v, x') \leq 6\mathcal{D}$.*

Then, for $\epsilon \in o(1/\log \tau)$ there is an algorithm that computes a series of connected clusters $\mathcal{K} = K_1, \dots, K_N$ with strong diameter $4(1+\epsilon)\mathcal{D}$ where for all nodes $v \in V$ and all $\epsilon \leq \gamma \leq \frac{1}{32}$, it holds:

$$\mathbf{Pr}[B(v, \gamma\mathcal{D}) \subset K(v)] \geq e^{-\Theta((\gamma+\epsilon)\log \tau)} - O(1/n^c).$$

Here, $K(v)$ denotes the cluster that contains v. The algorithm can be implemented with one $(1+\epsilon)$ approximate SetSSP computation and $\tilde{O}(1)$ minor aggregations.

Technically, this algorithm is a generalization of the algorithm in [32] that is based on **exact** shortest path computations. This algorithm is itself derived from [50]. Our algorithm replaces all these exact computations through $(1+\epsilon)$-approximate calculations. The main analytical challenge is carrying the resulting error ϵ through the analysis to obtain the bounds in the theorem. The same approach was already used by Becker, Lenzen, and Emek [13]. However, our analysis is more fine-grained w.r.t. to the impact of the approximation parameter ϵ.

In the following, we give the high-level idea behind the construction. An intuitive way to think about the clustering process from [32, 50] is as follows: Each center x draws value $\delta_x \sim \mathcal{D} \cdot \mathsf{Texp}(2 + 2\log \tau)$ and wakes up at time $\mathcal{D} - \delta_x$. Then, it begins to broadcast its identifier. The spread of all centers is done in the same unit tempo. A node v joins the cluster of the *first* identifier that reaches it, breaking ties consistently. If we had $O(\mathcal{D})$ time, we could indeed implement it exactly like this. However, this approach is infeasible for a general $\mathcal{D} \in \tilde{\Omega}(1)$ in weighted graphs as $\mathcal{D}$ may be arbitrarily large. Instead, we will model this intuition using a virtual super source s and shortest path computations. For each center $x \in \mathcal{X}$, we independently draw a value $\delta_x \sim \mathcal{D} \cdot \mathsf{Texp}(2 + 2\log \tau)$ from the truncated exponential distribution with parameter $2 + 2\log(\tau)$. We assume that τ (or some upper bound thereof) is known to each node. Then, we add a virtual source s with a weighted virtual edge (s, x) to each center $x \in \mathcal{X}$ with weight $w_x := (\mathcal{D} - \delta_x)$. Then, we compute a $(1+\epsilon)$-approximate SetSSP from s with $\epsilon \leq \frac{1}{40 \log \tau}$. Any node joins the cluster of the *last* center on its (approximate) shortest path to s.

With exact shortest paths, this construction would preserve the intuition. Any node whose shortest path to s contains center x as its last center on the path to s would have been reached by x's broadcast first. The statement is not that simple with approximate distances as the approximation may introduce a *detour*, so the center that minimizes $(\mathcal{D} - \delta_x) + d(x, v)$ may not actually cluster a node $v \in V$. In particular, two endpoints of a short edge $\{v, w\}$ may end up in different clusters although the same center minimizes the distance to both. The approximation error for v could be large, while for w, it is very low, which can cause undesired behavior. The nodes will only be added to the same cluster, if its center's head start (which is determined by the random distance to the source) is large enough to compensate for the *approximation errors*. This, however, depends on the approximation and *not* an the length of the edge we consider. Nevertheless, if the error ϵ is small enough, we get similar clustering guarantees that are good enough for our purposes. In particular, if the difference in distance between the two closest centers is larger than $O(\epsilon\mathcal{D})$, the clustering behaves very similar to its counterpart with exact distances. In our analysis, we show this by carefully carrying the approximation error through the analysis. We do not introduce any fundamental new techniques, but simply carry the approximation error all the way through the analysis of the corresponding clustering process with exact distances presented in [32].

In addition to Theorem 8 itself, this also allows us to observe the positive correlation between nodes that are on the shortest path to some cluster center. This closer look also allows us to prove the following technical fact about the pseudo-padded decompositions:

▶ **Lemma 9** (Well-Separated Nodes). *Let $\mathcal{K}$ be a partition computed by the algorithm from Theorem 8. Then, there a constant $c \geq 1$ such that for given node $u \in K_i$ with probability $(1/2)$ the following holds: There is path $P_u := (u = v_1, \dots, v_N = x_i)$ of length at most $2(1+\frac{1}{40 \log \tau})\mathcal{D}$ to the cluster center x_i of K_i where for each $v_j \in P_u$, it holds $B\left(v_j, \frac{\mathcal{D}}{c \log \tau}\right) \subseteq K_i$.*

This lemma states that if a single node $v \in V$ is padded, all nodes on a short path to its cluster center are likely padded as well. While this initially sounds very technical, it roughly translates to the following: Consider a clustering $\mathcal{K} = K_1, \dots, K_N$. Then for a suitably chosen $\rho := \frac{\mathcal{D}}{c \cdot \log \tau}$, in each cluster $\mathcal{K}_i = (V_i, E_i)$ there is a constant fraction of nodes $V_i' \subseteq V_i$ in the distance ρ to their closest node in a neighboring cluster $K_j \neq K_i$ **and** for any two $v, w \in V_i'$ there is a path of length $6\mathcal{D}$ (via the cluster center) that only consists of nodes in V_i'. This insight will be *crucial* for our next algorithm. A more detailed description and the full proof of both lemmas can be found in the full version [24].

2.2 Strong LDCs from Pseudo-Padded Decompositions

In this section, we present a generic algorithm that creates an LDC with strong diameter of quality $(O(\log \tau), 1/2)$ if the number of nodes that can be centers of clusters has been sufficiently *sparsed out*. In particular, we suppose that each node can only be in one of τ clusters. For this case, we show that it holds:

▶ **Theorem 10** (A Generic Clustering Theorem). *Let $\mathcal{D} > 0$ be a distance parameter, $G := (V, E, \ell)$ a (possibly weighted) undirected graph, and $\mathcal{X} \subseteq V$ be a set of marked nodes. Suppose that for each node $v \in V$, the following two properties hold:*

- ***Covering Property:*** *There is at least one $x \in \mathcal{X}$ with $d(v, x) \leq \mathcal{D}$.*
- ***Packing Property:*** *There are at most τ centers $x' \in \mathcal{X}$ with $d(v, x') \leq 6\mathcal{D}$.*

Then, there is an algorithm that creates an LDC of strong diameter $8\mathcal{D}$ with quality $(O(\log \tau), 1/2)$. The algorithm can be implemented with $\tilde{O}(1)$ minor aggregations and $\tilde{O}(1)$ $(1+\epsilon)$-approximate SetSSP computations where $\epsilon \in O(1/\log^2 n)$.

The algorithm uses the so-called *blurry ball growing* (BBG) technique. This is perhaps the key technical result that Becker, Emek, and Lenzen introduced in [13]. It provides us with the following guarantees:

▶ **Lemma 11** (Blurry Ball Growing (BBG), cf. [13, 55]). *Given a subset $S \subseteq V$ and an arbitrary parameter $\rho > 0$, an algorithm for BBG outputs a superset $S' \supseteq S$ with the following properties*

1. *An edge $\{v, w\} \in E$ of length $\ell_{(v,w)}$ is cut with probability $\mathbf{Pr}[v \in S', w \notin S'] \leq O\left(\frac{\ell_{(v,w)}}{\rho}\right)$.*
2. *For each node $v \in S'$, it holds $d(v, S) \leq \frac{\rho}{1-\alpha} \leq 2\rho$ where $\alpha \in O(\log \log n / \log n)$.*

BBG can be implemented using $\tilde{O}(1)$ $(1+\epsilon)$ approximate SetSSP computations with $\epsilon \in O\left((\log \log n / \log n)^2\right)$.

This technique allows us to create clusters with a low probability of cutting an edge while only having access to approximate shortest paths. Note that in [55] the dependency on ϵ was improved to $O(1/\log n)$. The paper also presents a deterministic version of blurry ball growing. However, we will only use the probabilistic version introduced above.

Given this definition, we now give an overview of the algorithm that creates an LDC: In the following, define $\rho := \frac{\mathcal{D}}{c \cdot \log \tau}$ where c is the constant from Lemma 9. The algorithm first computes a pseudo-padded decomposition $\mathcal{K} = K_1, \dots, K_N$ of strong diameter $8\mathcal{D}$ using the algorithm of Theorem 8. To do this, we choose the given set $\mathcal{X}$ of marked nodes as the set of cluster centers.

The following steps are executed in each cluster K_i in parallel. Within each cluster K_i, we determine a so-called *inner cluster* $K_i' \subseteq K_i$ of strong diameter $6\mathcal{D}$ where each node $v \in V_i'$ has distance (at least) $\rho' := \rho/2$ to the closest node in different cluster $K_j \neq K_i$. These inner clusters can be determined via two (approximate) SetSSP computations: First, all nodes calculate the 2-approximate distance to the closest node in a different cluster. To do this, we first create a virtual supersource s, and all nodes $v \in K_i$ with an edge to a node $w \in K_j$ create a virtual edge of length $\ell_{(v,w)}$ to s. If they have several such neighbors, they choose the shortest edge. Then, we perform a 2-approximate SetSSP from s on the resulting virtual graph that consists of all edges within a cluster and the virtual edges to s. In other words, the edges $\{v, w\}$ between clusters are not considered.

We then mark all nodes where this 2-approximate distance exceeds $2\rho'$ as *active nodes*. Next, we consider the graph G' induced by the active nodes. Using a $(1+\frac{1}{40\log\tau})$-approximate SetSSP calculation, the active nodes compute if they have a path of length at most $3\mathcal{D}$ to their cluster center in G'. If so, they add themselves to the inner cluster K_i' of K_i. Recall that their exact distance to the next cluster is at least ρ' as the paths are 2-approximate. Further, for any two nodes $v, w \in K_i'$ there is a path of length at most $6\mathcal{D}$ via the cluster center and so the inner cluster has a strong diameter of $6\mathcal{D}$. Thus, this procedure *always* results in an inner cluster K_i' with the desired properties. In the third and final stage of the algorithm, the actual clusters are computed. To this end, the *inner clusters* $\mathcal{K}' := \bigcup_{i=1}^{N} K_i'$ are used as the input set to the BBG procedure from Lemma 11. In particular, we choose the parameter for the BBG to be $\rho'' := \rho'/2$ and compute the superset $S(\mathcal{K}') := \mathsf{blur}(\mathcal{K}', \rho'')$. Now define final clustering $\mathcal{C} = C_1, \dots, C_N$ where $C_i := K_i \cap S(\mathcal{K}')$. In other words, the cluster C_i contains the inner cluster K_i' and all nodes from K_i added by the BBG process.

The resulting clustering fulfills all three properties required by Theorem 10. For the cutting probability, note that the distance from an inner cluster to a different cluster is ρ', and BBG only adds nodes in distance smaller than $2\rho'' = \rho'$. Thus, all edges with an endpoint in cluster K_i are only cut by the blurring process and not by the initial pseudo-padded decomposition. Thus, the choice of ρ'' implies that edges are cut with correct probability of $O(\ell/\rho'') = O(\frac{\ell \cdot \log\tau}{\mathcal{D}})$ by Lemma 11. The (strong) diameter of the cluster is at most $8\mathcal{D}$ as each node is in the distance at most $\rho \leq \mathcal{D}$ to a node from an inner cluster, and all nodes in an inner cluster are in the distance $6\mathcal{D}$ to one another. Therefore, it only remains to show that a nodes is clustered with prob. $1/2$. We can prove this via Lemma 9. For all nodes $v \in V$ the following holds with probability $1/2$: Node $v \in K_i$ **and** all nodes on the path P_v of length $2(1+\frac{1}{40\log\tau})\mathcal{D}$ to its cluster center x_i are in distance at least ρ to the closest node in a different cluster. Therefore, all these nodes will mark themselves active. As the 2-approximate SetSSP only overestimates, these nodes compute a distance more than $\rho = 2\rho'$, which causes them to be active. Moreover, this implies that the path P_v is fully contained in the graph G' induced by active nodes. As the path P_v is of length $2(1+\frac{1}{40\log\tau})\mathcal{D}$ and ends in the cluster center x_i, the $(1+\frac{1}{40\log\tau})$-approximate SetSSP computation will find a path of length at most $(1+\frac{1}{40\log\tau})2(1+\frac{1}{40\log\tau})\mathcal{D} \leq 3\mathcal{D}$ in G'. Therefore, with a probability of at least $1/2$, node v and all nodes on the path to the cluster center will be added to the inner cluster K_i'. Thus, Theorem 10 follows as half of all nodes are in an inner cluster and are therefore guaranteed to be in some cluster C_i. A more detailed description and the full proof can be found in the full version [24].

2.3 Strong LDDs from Strong LDCs

In this section, we show that we can create an LDD with quality $O(\log n)$ for any graph within $\tilde{O}(1)$ minor aggregations and $(1+\epsilon)$-approximate SetSSP's. Note that, at first glance, the theorem follows almost directly from Theorem 10. Choose $\mathcal{X} = V$ and $\mathcal{D}' = 8\mathcal{D}$ and apply

the theorem to a graph G. We obtain a clustering with diameter $\mathcal{D}'$ if each node has one node in distance $\mathcal{D}'/8$ and at most τ nodes in distance $\mathcal{D}'$. Clearly, for every possible choice of parameter $\mathcal{D}'$, each node has at least one marked node in distance $\mathcal{D}'/8$, namely itself. Further, for every possible choice of parameter $\mathcal{D}'$, each node has at most n marked node in distance $\mathcal{D}$ because there are only n nodes in total. Thus, by choosing all nodes as centers, we obtain a clustering with quality $O(\log n)$ and a diameter we can choose freely. While the diameter and edge-cutting probability of the resulting clustering are both (asymptotically) correct, we can only guarantee that at least half of the nodes are clustered on expectation. To put it simply, we do not have a partition as required. However, we can reapply our algorithm to the graph induced by the *unclustered* nodes. This clusters half of the remaining nodes on expectation. Thus, by applying Markov's inequality, a constant fraction of nodes is clustered with constant probability. It is easy to see that if we continue this for $O(\log n)$ iterations, we obtain the desired partition, w.h.p. Further, this will not significantly affect the edge-cutting probability, as each edge that is not cut will be added to a cluster with constant probability. Therefore, there cannot be too many iterations where the edge can actually be cut. Thus, applying the algorithm until all nodes are clustered intuitively produces the required LDD of quality $O(\log n)$. We formalize and prove this intuition in the following lemma.

▶ **Lemma 12.** *Let $G = (V, E, \ell)$ be a weighted graph. Let $\mathcal{A}$ be an algorithm that for each subgraph $G' \subseteq G$ can create clusters $K_1, \dots, K_N$ where each edge is cut between clusters with probability at most α, and each node is added to a cluster with probability at least β. Then, recursively applying $\mathcal{A}$ to G until all its nodes are in a cluster creates an LDD of quality $O(\alpha\beta^{-2})$. The procedure requires $O(\beta^{-1} \log n)$ applications of $\mathcal{A}$, w.h.p.*

The full proof can be found in the full version [24]. Together, Lemma 12 and Theorem 10 imply Theorem 5. Note that we state Lemma 12 more generally than needed, so we can reuse it in the next section.

3 Weak Separators from Approximate Shortest Paths

We now move from arbitrary graphs to k-path separable graphs. This section presents our main technical result and considers the distributed computation of separators for k-path separable graphs. Recall that a separator S is a *subgraph* of G such that in the induced subgraph $G \setminus S$ that results from removing S from G, every connected component only contains a constant fraction of G's nodes. Separators are a central tool in designing algorithms for restricted graph classes as they give rise to efficient divide-and-conquer algorithms. Our algorithms are no exception. In particular, we will compute a novel type of separator as the main building blocks of our clustering result. Note that the distance is taken w.r.t. to G, which could be a subgraph of a larger graph. We formally define our weak separators as follows:

▶ **Definition 13** (Weak κ-Path $(\mathcal{D}, \epsilon)$-Separator)**.** *Let $G := (V, E, \ell)$ be a weighted graph, $\mathcal{D} > 1$ be an arbitrary distance parameter, and $\epsilon > 0$ be an approximation parameter. Then, we call the set $S(\mathcal{D}, \epsilon) := (\mathcal{P}_1, \dots, \mathcal{P}_\kappa)$ with $\mathcal{P}_i := (P_i, B_i)$ a weak κ-path separator, if it holds:*

1. *Each $P_i \in \mathcal{P}_i$ is a (approximate) shortest path in $G \setminus \bigcup_{j=1}^{i-1} \mathcal{P}_j$ of length at most $4\mathcal{D}$.*
2. *Each $B_i \subseteq B_G(P_i, \epsilon\mathcal{D})$ is a set of nodes surrounding path P_i.*
3. *For all $v \in (V \setminus \bigcup_{j=1}^{\kappa} \mathcal{P}_i)$ it holds $|B_{G\setminus S}(v, \mathcal{D})| \leq (7/8) \cdot n$.*

Note that this definition is generic enough to also include vanilla k-path separators (where each set B_i only contains the path itself and no further nodes) and *traditional* vertex separators (where each path is a single node). Thus, for certain k-path separable graphs, we can already use existing algorithms to compute these separators in distributed and parallel models[5]. For general k-path separable graphs, however, we need new techniques.

Why Computing Separators for k-Path Separable Graphs is Hard

Before we go into details, let us first review some background on separators that motivates why we will mainly work with the weaker concept. To this end, we first review prior results and identify possible difficulties. Starting with positive results, it *is* possible to construct path separators for planar graphs in a distributed setting. In fact, the state-of-the-art algorithms for computing DFS trees in [39] or computing a graph's (weighted) diameter in [48] in planar graphs construct a path separator for planar graphs. The algorithm presented in these papers constructs a path separator in $\tilde{O}(\text{HD})$ time, w.h.p. However, while the existence of an efficient algorithm for a planar graph gives hope, it turns out to be very tough for other k-path separable graphs. The constructions in both [39] and [48] heavily exploit the fact that a planar embedding of a (planar) graph can be computed in $\tilde{O}(\text{HD})$ time in CONGEST due to Ghaffari and Haeupler in [36]. Given a suitable embedding, the ideas could be generalized. Thus, finding a suitable embedding could be a way to extend this algorithm to more general k-separable graphs. A good candidate for such an embedding can be found in [2]. Abraham and Gavoille present an algorithm that computes a $f(r)$-path separator for K_r-free graphs [2]. Their algorithm relies on a complex graph embedding algorithm by Seymour and Robertson [54]. On a very high level, the embedding divides G into subgraphs that can almost be embedded on a surface similar to planar graph. By *almost*, we mean that in each subgraph, there is only small number of non-embeddable parts (with special properties) that need to be handled separately. The concrete number of these parts depends only on r. Given such an embedding, Abraham and Gavoille show that one can efficiently compute k-path separator. Sadly, we have little hope that the algorithm that computes the embedding can be efficiently implemented in a distributed (or even parallel) context. It already requires $n^{O(r)}$ time in the sequential setting. Moreover, it requires sophisticated techniques that can not be trivially sped up by running them in parallel. Thus, it remains elusive (to the authors at least) how to implement this algorithm in any distributed or even parallel model.

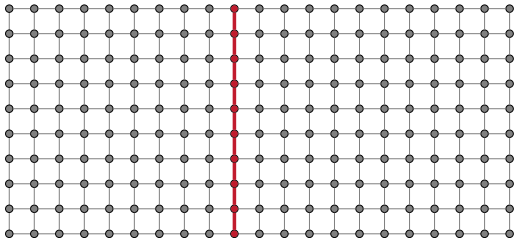
Why Computing Weak Separators for k-Path Separable Graphs is Easier

Now that we have established why it is difficult to compute k-path separators, we can proceed to the initially promised weak separators. Our main insight is that instead of computing an embedding in a distributed and parallel setting, we show that we get very similar guarantees using a weaker separator that can be computed without the embedding. Thus, we take a different approach to completely avoid the (potentially expensive) computation of an embedding. We strongly remark that, in doing so, we sacrifice some of the separator's useful properties. However, in our applications, this only results in some additional polylogarithmic factors in the runtime. To be precise, we prove the following lemma:

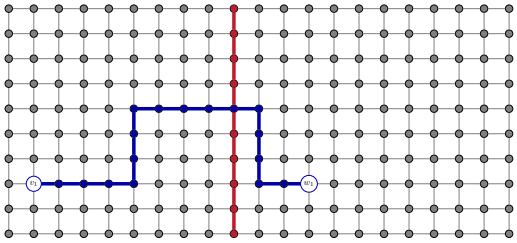
[5] For planar graphs, we can use the algorithm provided in [48] to compute a separator that consists of 4 paths. While for graphs of bounded tree-width τ, we can use [42] to obtain a separator that consists of τ nodes.

▶ **Theorem 14** (Weak Separators for k-Path Separable Graphs)**.** *Consider a weighted k-path separable graph $G := (V, E, \ell)$ with weighted diameter smaller than $\mathcal{D}$. For $\epsilon \geq 0$, there is an algorithm that constructs a weak $O(\epsilon^{-1} \cdot k \cdot \log n)$-path $(\mathcal{D}, \epsilon)$-separator, w.h.p. The algorithm uses $O(\epsilon^{-1} \cdot k \cdot \log n)$ minor aggregations and $O(\epsilon^{-1} \cdot k \cdot \log n)$ 2-approximate shortest path computations, w.h.p.*

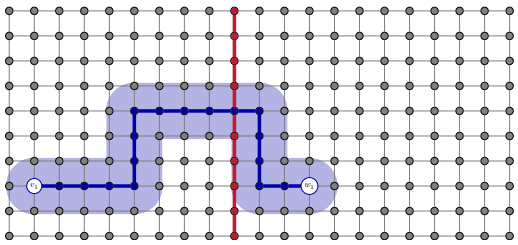
Surprisingly, the algorithm behind the lemma can be stated in just a few sentences. The core idea is to iteratively sample approximate shortest paths and nodes close to these paths and remove them from the graph until all remaining nodes have few nodes in distance $\mathcal{D}$, w.h.p. More precisely, we start with a graph $G_0 = G$. The algorithm proceeds in $T \in O(\epsilon^{-1} \cdot k \cdot \log n)$ steps where in step t, we consider graph $(V_t, E_t) := G_t \subset G_{t-1}$. A single step t works as follows: First, we pick a node $v \in V_t$ uniformly at random. Then, we compute a 2-approximate shortest path tree T_v from v. Let $W_v \subset V_t$ be set of all nodes in distance at most $4 \cdot \mathcal{D}$ to v in T_v. We pick a node $w \in W_v$ uniformly at random and consider the corresponding path $P_t = (v, \dots, w)$. Using P_t as a set-source, we compute a 2-approximate shortest path tree T_{P_t} from P_t. Now denote B_{P_t} to be the nodes in the distance at most $\epsilon \cdot \mathcal{D}$ to P_t in G (and **not** G_t). Finally, we remove P_t and B_{P_t} from G_t to obtain G_{t+1}. Note that, due to the approximation guarantee, this removes all nodes in distance at least $\epsilon/2 \cdot \mathcal{D}$ from P_t in G. Further, all steps can be computed using 2-approximate paths (and few aggregation to pick the random nodes) and are completely devoid of any complex embedding. A visualization of the algorithm can be found in Figure 1.



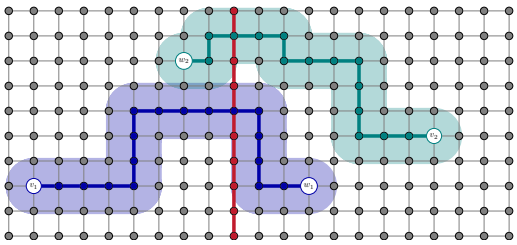
(a) Marked in red is a separator path unknown to the algorithm.



(b) When sampling a random path of bounded length, the probability of crossing the separator is constant.



(c) We remove nodes in distance $\epsilon\mathcal{D}$ around the path. This also removes an $\epsilon\mathcal{D}$-fraction of the separator path.



(d) If the residual graph is not well-separated, the probability of crossing (what remains of) the separator is still constant.

Figure 1 We choose a two-dimensional mesh for illustration. While the operations in Figures **(b)** and **(c)** are straightforward to prove, the core of our analysis we will be the claim we make in Figure **(d)**.

Why does this work? As our main analytical tool, we consider the following potential:

$$\Phi_t := \left|\left\{v \in V_t \;\middle|\; |B_{G_t}(v, 2\mathcal{D})| \geq (7/8) \cdot n\right\}\right|$$

Further, for brevity, we write $\Phi_t(v) = 1$ if and only if $|B_{G_t}(v, 2\mathcal{D})| \geq (7/8) \cdot n$ and $\Phi_t(v) = 0$, otherwise. Note that $\Phi_t := \sum_{v \in V} \Phi_t(v)$.

We make the following important observation: As soon as this potential drops below $(7/8) \cdot n$, we have more than $(1/8) \cdot n$ nodes with less than $(7/8) \cdot n$ nodes in distance $2\mathcal{D}$. This implies that no node can have more than $(7/8) \cdot n$ nodes in distance $\mathcal{D}$. Suppose for contradiction that there is a node $v \in V_t$ with more than $(7/8) \cdot n$ nodes in distance $\mathcal{D}$. By the triangle inequality, all these nodes would be in distance $2\mathcal{D}$ to each other. Thus, the potential would be bigger than $(7/8) \cdot n$ as there are more than $(7/8) \cdot n$ nodes with more than $(7/8) \cdot n$ nodes in distance $\mathcal{D}$. This is a contradiction and therefore the nodes sampled until this point are the desired weak separator.

Therefore, we want to bound the number of steps until the potential drops. For this, we will first show the following useful fact about k-path separators, namely

▶ **Lemma 15.** *Let $G := (V, E, w)$ be a weighted k-path separable graph of n nodes with (weighted) diameter $\mathcal{D}$. Then, there exists a set $\mathcal{B} = \{P_1, \ldots, P_\kappa\}$ with $\kappa \leq k$ simple paths of length at most $32\mathcal{D}$ such that for all $v \in V$, it holds $B_{G \setminus \mathcal{B}}(v, 4\mathcal{D}) \leq (3/4) \cdot n$.*

In other words, a subset of bounded length paths intersects with a constant fraction of all paths our algorithm can potentially sample. The proof requires copious use of the triangle inequality and the pigeonhole principle and can be found in the full version [24].

Now, consider the event that the algorithm samples a path P_t that *crosses* some path of $\mathcal{B}$, i.e., the path we sample contains (at least) one node from one of the κ paths in $\mathcal{B}$. In this case, we remove a subpath of length $(\epsilon/2)\mathcal{D}$ from this path when we remove the set B_{P_t} around the P_t. This follows because we determine the set B_t with respect to the original graph G and not G_t. Thus, after sampling $O(\epsilon^{-1} \cdot \mathcal{D})$ paths that cross $\mathcal{B}$, we must have completely removed $\mathcal{B}$ from G. If $\mathcal{B}$ is removed, each node has at most $(3/4) \cdot n$ nodes in distance $\mathcal{D} \leq 4\mathcal{D}$ per definition and we are done. Thus, we will bound the time until **either** the potential drops **or** a total of $O(\epsilon^{-1} \cdot k)$ paths cross $\mathcal{B}$. Recall that in each step t, we sample a path from v_t to w_t. In the following, denote the event that the path from v_t to w_t crosses $\mathcal{B}$ as $v_t \rightsquigarrow_{\mathcal{B}} w_t$. By definition of $\mathcal{B}$, for each node $v \in V$, the set of nodes where all paths of length smaller than $4 \cdot \mathcal{D}$ contain a node of $\mathcal{B}$ is of size at least $(1/4) \cdot n$. Thus, for all nodes $v \in V_t$ with $\Phi_t(v) = 1$, at least $(1/4) \cdot n - (1/8) \cdot n \geq (1/8) \cdot n$ of these nodes must still be in distance $2\mathcal{D}$. All these nodes might be chosen as the path's endpoint P_t if we pick v as the starting point as the 2-approximate distance is at most $4\mathcal{D}$. So, if we sample one of them, we cross $\mathcal{B}$ as all paths of length at most $4\mathcal{D}$ cross $\mathcal{B}$. Thus, in a configuration with potential $\Phi_t \geq (7/8) \cdot n$, the probability to sample a path that crosses $\mathcal{B}$ is at least

$$\begin{aligned}\mathbf{Pr}[v_t \rightsquigarrow_{\mathcal{B}} w_t \mid \Phi_t \geq (7/8) \cdot n] &\geq \mathbf{Pr}[\Phi_t(v_t) = 1 \mid \Phi_t \geq (7/8) \cdot n] \cdot \mathbf{Pr}[v_t \rightsquigarrow_{\mathcal{B}} w_t \mid \Phi_t(v_t) = 1] \\ &\geq \frac{1}{n} \cdot \Phi_t \cdot \frac{1}{n} \cdot (1/8) \cdot n \geq \frac{7}{64} \geq \frac{1}{16}\end{aligned}$$

Thus, if the potential has not dropped after T steps, the expected number of crossings is $(1/16) \cdot T$. Using standard Chernoff-like tail estimates, we can show that for any $T \in \Omega(\log n)$, this would imply that at least $\frac{T}{32}$ paths crossed $\mathcal{B}$, w.h.p. Thus, within $T \in O(\epsilon^{-1} \cdot k \cdot \log n)$ steps, we either reached a step with low potential or had sufficiently many crossings to remove $\mathcal{B}$. In either case, we are done as each node had less than $7/8 \cdot n$ nodes in distance $\mathcal{D}$. As up until this point we sampled $T \in O(\epsilon^{-1} \cdot k \cdot \log n)$ paths of length $4\mathcal{D}$ and all nodes in distance $\epsilon\mathcal{D}$ around these paths, this set must be the desired weak separator.

4 Low-Diameter Decompositions for k-path Separable Graphs

In this section, we prove Theorem 6. Our algorithm will combine the insights we gathered for LDD's and LDC's in general graphs with our novel separator construction. Our main technical result is the following proposition:

▶ **Proposition 16** (A Clustering Theorem for Restricted Graphs). *Let $\mathcal{D} > 0$ be an arbitrary distance parameter and $G := (V, E, \ell)$ be a (possibly weighted) $\tilde{O}(1)$-path separable graph. Then, there is an algorithm that creates an LDC of strong diameter $\mathcal{D}$ with quality $(O(\log k + \log\log n), 1/2)$. The algorithm can be implemented in $\tilde{O}(k)$ minor aggregations and $\tilde{O}(k)$ $(1+\epsilon)$-approximate SetSSP computations with $\epsilon \in O\left(1/\log^2 n\right)$.*

Crucially, the proposition is sufficient to prove Theorem 6 for $\tilde{O}(1)$-separable graphs: For $k \in \tilde{O}(1)$ one application of the algorithm creates a clustering with strong diameter $\mathcal{D}$ and cutting probability $O\left(\frac{\ell_z \cdot (\log\log n)}{\mathcal{D}}\right)$. Now recall that universally k-path separable graphs are closed under minor-taking. Therefore, the graph $G \setminus \mathcal{K}$ is also k-path separable, and we can apply the algorithm to $G \setminus \mathcal{K}$ with the same guarantees to cluster at least half of the remaining nodes. By Lemma 12, after $O(\log n)$ recursive applications, all nodes are clustered, and we obtain an LDD with quality $O(\log\log n)$ as required. Further, for $k \in \tilde{O}(1)$ one iteration of the algorithm can, w.h.p., be implemented in $\tilde{O}(1)$ minor aggregations and $\tilde{O}(1)$ $(1+\epsilon)$-approximate SetSSP computations with $\epsilon \in O\left(1/\log^2 n\right)$. Thus, $O(\log n)$ recursive applications are within the required time complexity of Theorem 6.

Thus, for the remainder, we will focus on Proposition 16. From a high-level perspective, the algorithm behind Proposition 16 proceeds in two distinct phases: the *backbone phase* and the *refinement phase*. In the backbone phase, we create a special partition $\mathcal{K}[\mathcal{B}] := K(\mathcal{B}_1), \dots, K(\mathcal{B}_N)$ of G. Each node $v \in K(\mathcal{B}_i)$ is in distance at most $\mathcal{D}_{BC} < \mathcal{D}$ to some subset $\mathcal{B}_i \subset K(\mathcal{B}_i)$ that consist of $\tilde{O}(k)$ (possibly disjoint) paths of bounded length of $O\left(\mathcal{D}\log^2\right)$. These paths may not necessarily be short**est** paths in G or even $K(\mathcal{B}_i)$, only their length is bounded. We call these subsets $\mathcal{B}_1, \dots, \mathcal{B}_N$ the *backbones* of the clusters. In contrast to an LDD, the clusters do **not** have simple nodes $v \in V$ as centers. As a result, the diameter of the resulting clusters might be much larger than $\mathcal{D}$. Moreover, a single cluster might not even be connected. In the second phase, we turn the resulting clustering into an LDC with connected clusters of strong diameter $\mathcal{D}$. We do so by computing an LDC $\mathcal{K}_i = K_{i_1}, \dots, K_{i_N}$ of strong diameter $\mathcal{D}$ in each cluster $K(\mathcal{B}_i)$. Here, we exploit the special structure of the backbone cluster and choose a carefully chosen subset of nodes from each backbone path as possible cluster centers such that no more than $\tilde{O}(k)$ centers can cluster a given node. These centers will then be used as an input to Theorem 10 to refine each backbone cluster to a set of connected clusters of strong diameter $\mathcal{D}$.

This algorithm is inspired by the works of Abraham, Gavoille, Gupta, Neiman, and Talwar [3, 4] and the subsequent improvement of their algorithm by Filtser [32]. Surprisingly, we do not require fundamental changes to obtain an efficient distributed algorithm. Both algorithms essentially follow the same two-phase structure sketched above. They first constructed a partition around paths sampled in a certain way, then refined each partition individually to obtain proper clusters centered around nodes. For reference, Abraham, Gavoille, Gupta, Neiman, and Talwar dubbed these subsets around which the partitions are constructed *skeletons* while Filtser used the term *r-core*. However, their key properties are (almost) the same. We use the different term *backbones* because we construct them slightly differently. Our contribution when comparing our algorithm to [3, 4, 32] is that we show a) that the exact shortest path can be replaced by approximate paths and the blurry ball growing procedure

from Lemma 11 and, therefore, can be implemented efficiently and b) that we can use our weak separator from Section 3 to parallelize the algorithm and achieve a logarithmic recursion depth efficiently.

In the remainder of this section, we sketch the backbone phase in Subsection 4.1 and the refinement phase in Subsection 4.2. Together, the lemmas presented in these sections will prove Proposition 16.

4.1 The Backbone Phase

In this section, we sketch the first phase of the algorithm. We begin by giving a general definition of the special type of clustering computed in this phase. Formally, we define it as follows:

▶ **Definition 17** (Backbone Clustering). *Let $\mathcal{D} > 0$ be a distance parameter and $G := (V, E, \ell)$ a (possibly weighted) graph. Then, a (α, β, κ)-backbone clustering is a series of disjoint subgraphs $K(\mathcal{B}_1), \dots, K(\mathcal{B}_N)$ with $K(\mathcal{B}_i) = (V_i, E_i)$ where $V_1 \sqcup \dots \sqcup V_N = V$. Further, the following three conditions hold:*

1. *Each node $v \in K(\mathcal{B}_i)$ is in distance at most $\mathcal{D}$ to $\mathcal{B}_i$.*
2. *An edge of length ℓ is cut between clusters with probability $O\left(\frac{\ell \cdot \alpha}{\mathcal{D}}\right)$.*
3. *Each backbone $\mathcal{B}_i$ consists of at most κ paths of length $O(\beta\mathcal{D})$.*

Given this definition, we show that we can use our algorithm for constructing weak separators to create a good backbone clustering of a graph G. To be precise, we show the following lemma:

▶ **Lemma 18** (Backbone Clustering). *Let $\mathcal{D} > 0$ be a distance parameter and $G := (V, E, \ell)$ a (possibly weighted) k-path separable graph. Then, there is an algorithm that creates an (α, β, κ)-backbone clustering with $\alpha \in O(\log(k \log n))$, $\beta \in O(\log^2 n)$, $\kappa \in O(k \log^2 n)$ and pseudo-diameter $\mathcal{D}$ for G. The algorithm can be implemented with $\tilde{O}(k)$ minor aggregations and $(1+\epsilon)$-approximate SetSSP computations where $\epsilon \in O\left(1/\log^2 n\right)$.*

The algorithm works as follows: Let $G_t \subset G$ be the graph at the beginning of the t^{th} recursive step, where initially $G_0 = G$. At the beginning of each recursive step, we create an LDD $C_1, \dots, C_{N'}$ of diameter $\mathcal{D}' \in O(\mathcal{D} \log^2 n)$ of G_t. We use the algorithm from Theorem 5 for this. The edges between two clusters C_i and C_j are removed and will not be considered in the following iterations. We denote the graph that results from removing all these edges as G'_t. Then, in each C_i, we compute a weak $\tilde{O}(k)$-path $(\mathcal{D}', \epsilon)$-separator $\mathcal{S}_i$ with $\epsilon \in O(1/\log^2 n)$ using the algorithm from Theorem 14. Recall that we defined this separator to be the union of $O(\epsilon^{-1} \cdot k)$ paths of length $4 \cdot \mathcal{D}'$ and some nodes in distance $\epsilon\mathcal{D}'$ to these paths. The $O(\epsilon^{-1} \cdot k)$ paths in $\mathcal{S}_i$ will be the backbone of a cluster. To add further nodes to the cluster, we proceed in two steps: First, we draw $X_i \sim \mathsf{Texp}(4 \log(\log n))$ and compute a $(1+\epsilon)$-approximate SetSSP from $\mathcal{S}_i$. Then, we mark all nodes at a distance at most $(1+\epsilon)X_i \cdot \mathcal{D}_{BC}/4$ from $\mathcal{S}_i$. We denote this set as $K_{\mathsf{Texp}}(\mathcal{S}_i)$. Then, we apply the BBG procedure of Lemma 11 with parameter $\rho_{\mathsf{blur}} := \mathcal{D}_{BC}/c_{\mathsf{blur}} \cdot \log\log n$ from $K_{\mathsf{Texp}}(\mathcal{S}_i)$. Here, $c_{\mathsf{blur}} > 8$ is a suitably chosen constant. This results in a superset $K_{\mathsf{blur}}(\mathcal{S}_i) \subseteq C_i$ for each subgraph C_i. Finally, we add all sets $K_{\mathsf{blur}}(\mathcal{S}_i)$ from all clusters C_i to our clustering $\mathcal{K}[\mathcal{B}]$ and remove them from the graph.

We repeat this process with the remaining graph $G_{t+1} := G'_t \setminus \bigcup K_{\mathsf{blur}}(\mathcal{S}_i)$ until all subgraphs are empty. As we remove a weak $\mathcal{D}'$-separator in each step **and** then create an LDD with strong diameter $\mathcal{D}'$, the size of the largest connected components shrinks by a constant factor in each recursion. Thus, we require $O(\log n)$ recursions overall until all components are empty.

Recall that in each cluster $K_{\mathsf{blur}}(\mathcal{S}_i)$, we define the short paths $P_1, \dots, P_\kappa$ in the separator $\mathcal{S}_i$ to be the backbone. As there are $\kappa \in O(k \cdot \log^2 n)$ paths, it remains to analyze its pseudo-diameter and edge-cutting probability to prove that the resulting clustering is a proper backbone clustering. For an appropriate choice of $\epsilon \in O(1/\log^2 n)$, all nodes in $\mathcal{S}_i$ are in distance $\epsilon\mathcal{D}' \leq \mathcal{D}_{BC}/4$ to any path. Further as $(1+\epsilon) \leq 2$ and $X_i \cdot \mathcal{D}_{BC}/4 \leq \mathcal{D}_{BC}/4$, all nodes in $K_{\mathsf{Texp}}(\mathcal{S}_i)$ are in distance $(1+\epsilon) \cdot X_i \cdot \mathcal{D}_{BC}/4 \leq \mathcal{D}_{BC}/2$ to $\mathcal{S}_i$. Finally, the distance of any node in $K_{\mathsf{blur}}(\mathcal{S}_i)$ to any node in $K_{\mathsf{Texp}}(\mathcal{S}_i)$ is at most $2\rho_{\mathsf{blur}} \leq \mathcal{D}_{BC}/4$. Thus, the distance from each node $u \in K_{\mathsf{blur}}(\mathcal{S}_i)$ to any of the paths in the separator is at most

$$\begin{aligned} d(u, \mathcal{B}_i) &\leq d(v, K_{\mathsf{Texp}}(S_i)) + \mathbf{max}_{v \in K_{\mathsf{Texp}}(S_i)} d(v, S_i) + \mathbf{max}_{w \in S_i} d(w, \mathcal{B}_i) \\ &\leq \mathcal{D}_{BC}/4 + 2 \cdot \mathcal{D}_{BC}/4 + \mathcal{D}_{BC}/4 = 4\mathcal{D}_{BC}/4 = \mathcal{D}_{BC} \end{aligned}$$

Thus, only the edge-cutting probability remains to be shown. For each edge $z \in E$, two operations can cut an edge in each iteration. The LDD from Theorem 5 and the BBG procedure from Lemma 11. First, consider the LDD. As we pick $\mathcal{D}' \in O(\mathcal{D} \log^2 n)$, a single application of Theorem 5 cuts an edge with probability $O\left(\ell_z \log n / \mathcal{D}'\right) = O\left(\ell_z / \mathcal{D} \log n\right)$. Thus, as we apply it at most $O(\log n)$ times, the probability sums up to $O\left(\ell_z / \mathcal{D}\right)$ by the union bound. The other possibility is that z is cut by the BBG. By our choice of ρ_{blur}, the probability for the ball to cut z is $O\left(\ell_z \log(k \log n) / \mathcal{D}\right)$. Therefore, we cannot simply use the union bound to sum up the probabilities over all $O(\log n)$ iterations. Here, the intermediate step where we construct $K_{\mathsf{Texp}}(\mathcal{S}_i)$ comes into play. We can exploit that $K_{\mathsf{Texp}}(\mathcal{S}_i)$ must be *close* to either endpoint of z, i.e., within distance at most $2\rho_{\mathsf{blur}}$, in order for $\mathsf{blur}(K_{\mathsf{Texp}}(\mathcal{S}_i), \rho_{\mathsf{blur}})$ to cut it. The distance between $\mathsf{Texp}(\mathcal{S}_i)$ and z is determined by the exponentially distributed variable $X_i \cdot \mathcal{D}_{BC}/4$. Using the properties of the truncated exponential distribution, we can show that the probability of $\mathsf{Texp}(\mathcal{S}_i)$ being close to z without adding the endpoints of z to $\mathsf{Texp}(\mathcal{S}_i)$ is constant for our choice of parameters. Thus, the edge is safe from ever being cut before the blur procedure is even executed. This follows from the properties of the truncated exponential distribution. Thus, on expectation, there will only be a constant number of tries before z is either cut or safe. As we will see in the full analysis, this is sufficient for our probability bound.

Finally, verifying that each algorithm step requires $\tilde{O}(k)$ approximate SetSSP computations and minor aggregations is easy. We use three subroutines from Theorem 5, Theorem 14, and Lemma 11, which all require at most $\tilde{O}(k)$ approximate SetSSP computations and minor aggregations on a k-path separable graph. In addition, the algorithm only uses one more SetSSP; thus, the runtime follows. A more detailed description and the full proof can be found in the full version [24].

4.2 The Refinement Phase

For the second phase, we show a generic lemma that allows us to turn any backbone clustering as described in Definition 17 into an LDC. Thus, this statement is independent of the algorithm that creates the backbone clustering. To be precise, it holds:

▶ **Lemma 19.** *Suppose we have an algorithm $\mathcal{A}$ that creates a (α, β, κ)-backbone clustering with pseudo-diameter $\mathcal{D}_{BC}$ of a weighted graph G. Then, there is an algorithm that creates an LDC of strong diameter $16 \cdot \mathcal{D}_{BC}$ with quality $(O(\alpha + \log \kappa\beta), 1/2)$. The algorithm requires one application of $\mathcal{A}$, $\tilde{O}(\kappa)$ minor aggregations, and $\tilde{O}(1)$ approximate SetSSP computations with $\epsilon \in O(1/\log^2 n)$.*

The algorithm begins by constructing backbone clustering $K(B_1), \dots, K(B_N)$ using algorithm $\mathcal{A}$. The following steps are executed in parallel in each cluster $K(\mathcal{B}_i)$: On each path $P \in \mathcal{B}_i$ in the backbone, mark a subset of nodes in distance (at most) $\mathcal{D}_{BC}$ to each other and use

them as potential cluster centers. This can be done with one shortest path computation and two aggregations on each path: First, we let each node compute the distance to the first node of the path $v_1 \in P$. A simple SetSSP computation can do this; as there is only one path, the algorithm returns an exact result. Then, we let each node locally compute its *distance class* $\lceil \frac{d_P(v,v_1)}{\mathcal{D}_{BC}} \rceil$. Finally, we mark the first node in each distance class. A node can determine whether it is the first node by aggregating the distance class of both its neighbors in two aggregations. Having marked the nodes, we use them as centers $\mathcal{X}_i$ in the algorithm of Theorem 10 and compute an LDC in $K(\mathcal{B}_i)$. To determine the gurantees of this clustering, we need to determine the *covering* and *packing* properties of these centers. Each node $v \in K(\mathcal{B}_i)$ has at least one marked node in distance $2\mathcal{D}_{BC}$ by construction: It must have a path $P_j \in \mathcal{B}_i$ in distance $\mathcal{D}_{BC}$ and the next marked nodes on the path can also only be in distance $\mathcal{D}_{BC}$ to the node closest to v. Further, as a path of length $\beta\mathcal{D}_{BC}$ has β distance classes, we mark at most β nodes per path. As we limit the number of paths in $\mathcal{B}_i$ to κ and the length of each path to $\beta\mathcal{D}_{BC}$, the total number of centers is limited to $\beta \cdot \kappa$. Thus, each node in $K(\mathcal{B}_i)$ is covered by one center in distance $2\mathcal{D}_{BC}$ and *packed* by at most $\beta \cdot \kappa$ centers in distance $6 \cdot (2\mathcal{D}_{BC})$. Following Theorem 10, this gives us an LDC with strong diameter $8 \cdot (2\mathcal{D}_{BC})$ and quality $O(\log \beta\kappa)$. Finally recall that an edge can either be cut by the initial construction of the backbone clustering or by this procedure. By the union bound, an edge is cut with probability $O(\frac{\alpha+\log \beta\kappa}{\mathcal{D}})$ by either stage. As the algorithm consists of computing a backbone clustering, one SetSSP and one minor aggregations in each path to mark the nodes, and an application of Theorem 10, the proclaimed runtime follows.

Combing this result with Lemma 18, this creates the desired LDC with strong diameter $16\mathcal{D}_{BC}$ of quality $O(\log k + \log\log n)$ in each cluster. In particular, an edge is cut with probability $O(\frac{\ell \cdot \log k \log n}{\mathcal{D}})$ in either of the two phases. This proves the Proposition 16 when choosing $\mathcal{D}_{BC} = \mathcal{D}/16$. A more detailed description and the full proof can be found in the full version [24].

5 Conclusion & Future Work

We presented two novel distributed decomposition schemes for arbitrary and restricted graphs that perform favorably in all relevant parameters and apply to various models. First, for arbitrary graphs, we provided a decomposition scheme that produces decompositions with strong diameter and quality $O(\log n)$. The open question remains whether our algorithm's guarantees can also be achieved deterministically. As our results are deeply connected with the properties of the (truncated) exponential distribution, this likely requires very different techniques and/or novel ideas for derandomization. Second, we presented a decomposition scheme with quality $O(\log k + \log\log n)$ for universally k-path separable graphs, a graphs class containing many practical, relevant graphs. Albeit having a decomposition quality that is exponentially smaller than the quality for general graphs, our algorithm is *not* optimal. First of all, the dependency on n could be improved in future work. Further, for specific k-path separable graphs, the dependency on k could be problematic. For example, in a K_r-free graphs, we have $k = f(r)$ where $f(r)$ depends only on r. We note that $f(r)$, albeit being a constant for a constant r, is extremely large. Its concrete value is based on the Structure Theorem of Seymour and Robinson [54]. Their analysis is quite involved and stretches over a vast series of papers and articles. Moreover, due to its complexity, it is hard even to find an asymptotic characterization of that number. In [49], it is described as a "*tower of powers of* 2 *of height at most* 5*, with* r^{1000} *on top*". We emphasize that this number very likely is too high for practical applications and, therefore, further investigation is required.

References

1 Ittai Abraham, Yair Bartal, and Ofer Neiman. Nearly tight low stretch spanning trees. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 781–790. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.62.

2 Ittai Abraham and Cyril Gavoille. Object location using path separators. In Eric Ruppert and Dahlia Malkhi, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006, Denver, CO, USA, July 23-26, 2006*, pages 188–197. ACM, 2006. doi:10.1145/1146381.1146411.

3 Ittai Abraham, Cyril Gavoille, Anupam Gupta, Ofer Neiman, and Kunal Talwar. Cops, robbers, and threatening skeletons: padded decomposition for minor-free graphs. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 79–88. ACM, 2014. doi:10.1145/2591796.2591849.

4 Ittai Abraham, Cyril Gavoille, Anupam Gupta, Ofer Neiman, and Kunal Talwar. Cops, robbers, and threatening skeletons: Padded decomposition for minor-free graphs. *SIAM J. Comput.*, 48(3):1120–1145, 2019. doi:10.1137/17M1112406.

5 Ittai Abraham and Ofer Neiman. Using petal-decompositions to build a low stretch spanning tree. *SIAM J. Comput.*, 48(2):227–248, 2019. doi:10.1137/17M1115575.

6 Ioannis Anagnostides, Christoph Lenzen, Bernhard Haeupler, Goran Zuzic, and Themis Gouleakis. Almost universally optimal distributed laplacian solvers via low-congestion shortcuts. *Distributed Comput.*, 36(4):475–499, 2023. doi:10.1007/S00446-023-00454-0.

7 John Augustine, Kristian Hinnenthal, Fabian Kuhn, Christian Scheideler, and Philipp Schneider. Shortest paths in a hybrid network model. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1280–1299. SIAM, 2020. doi:10.1137/1.9781611975994.78.

8 Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 184–193. IEEE Computer Society, 1996. doi:10.1109/SFCS.1996.548477.

9 Yair Bartal. On approximating arbitrary metrices by tree metrics. In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 161–168. ACM, 1998. doi:10.1145/276698.276725.

10 Yair Bartal. Graph decomposition lemmas and their role in metric embedding methods. In Susanne Albers and Tomasz Radzik, editors, *Algorithms - ESA 2004, 12th Annual European Symposium, Bergen, Norway, September 14-17, 2004, Proceedings*, volume 3221 of *Lecture Notes in Computer Science*, pages 89–97. Springer, 2004. doi:10.1007/978-3-540-30140-0_10.

11 Surender Baswana and Sandeep Sen. A simple and linear time randomized algorithm for computing sparse spanners in weighted graphs. *Random Structures & Algorithms*, 30(4):532–563, 2007. doi:10.1002/RSA.20130.

12 Ruben Becker, Yuval Emek, Mohsen Ghaffari, and Christoph Lenzen. Distributed algorithms for low stretch spanning trees. In Jukka Suomela, editor, *33rd International Symposium on Distributed Computing, DISC 2019, October 14-18, 2019, Budapest, Hungary*, volume 146 of *LIPIcs*, pages 4:1–4:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.DISC.2019.4.

13 Ruben Becker, Yuval Emek, and Christoph Lenzen. Low Diameter Graph Decompositions by Approximate Distance Computation. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 50:1–50:29, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2020.50.

14 Guy E. Blelloch, Anupam Gupta, Ioannis Koutis, Gary L. Miller, Richard Peng, and Kanat Tangwongsan. Near linear-work parallel sdd solvers, low-diameter decomposition, and low-stretch subgraphs. In *SPAA 2011: Proceedings of the 23rd Annual ACM Symposium on Parallelism in Algorithms and Architectures, San Jose, CA, USA, June 4-6, 2011 (Co-located with FCRC 2011)*, SPAA '11, pages 13–22, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/1989493.1989496.

15 Jannik Castenow, Christina Kolb, and Christian Scheideler. A bounding box overlay for competitive routing in hybrid communication networks. In *Structural Information and Communication Complexity - 26th International Colloquium, SIROCCO 2019, L'Aquila, Italy, July 1-4, 2019, Proceedings*, pages 345–348, 2019. doi:10.1007/978-3-030-24922-9_26.

16 Yi-Jun Chang. Efficient distributed decomposition and routing algorithms in minor-free networks and their applications. In Rotem Oshman, Alexandre Nolin, Magnús M. Halldórsson, and Alkida Balliu, editors, *Proceedings of the 2023 ACM Symposium on Principles of Distributed Computing, PODC 2023, Orlando, FL, USA, June 19-23, 2023*, pages 55–66. ACM, 2023. doi:10.1145/3583668.3594604.

17 Yi-Jun Chang and Hsin-Hao Su. Narrowing the LOCAL-CONGEST gaps in sparse networks via expander decompositions. In Alessia Milani and Philipp Woelfel, editors, *PODC '22: ACM Symposium on Principles of Distributed Computing, Salerno, Italy, July 25 - 29, 2022*, pages 301–312. ACM, 2022. doi:10.1145/3519270.3538423.

18 Sam Coy, Artur Czumaj, Christian Scheideler, Philipp Schneider, and Julian Werthmann. Routing schemes for hybrid communication networks. In Sergio Rajsbaum, Alkida Balliu, Joshua J. Daymude, and Dennis Olivetti, editors, *Structural Information and Communication Complexity - 30th International Colloquium, SIROCCO 2023, Alcalá de Henares, Spain, June 6-9, 2023, Proceedings*, volume 13892 of *Lecture Notes in Computer Science*, pages 317–338. Springer, 2023. doi:10.1007/978-3-031-32733-9_14.

19 Andrzej Czygrinow, Michal Hanckowiak, and Wojciech Wawrzyniak. Fast distributed approximations in planar graphs. In Gadi Taubenfeld, editor, *Distributed Computing, 22nd International Symposium, DISC 2008, Arcachon, France, September 22-24, 2008. Proceedings*, volume 5218 of *Lecture Notes in Computer Science*, pages 78–92. Springer, 2008. doi:10.1007/978-3-540-87779-0_6.

20 Tijn de Vos. Minimum cost flow in the CONGEST model. In Sergio Rajsbaum, Alkida Balliu, Joshua J. Daymude, and Dennis Olivetti, editors, *Structural Information and Communication Complexity - 30th International Colloquium, SIROCCO 2023, Alcalá de Henares, Spain, June 6-9, 2023, Proceedings*, volume 13892 of *Lecture Notes in Computer Science*, pages 406–426. Springer, 2023. doi:10.1007/978-3-031-32733-9_18.

21 Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, Heidelberg; New York, fourth edition, 2010.

22 Emilie Diot and Cyril Gavoille. Path separability of graphs. In Der-Tsai Lee, Danny Z. Chen, and Shi Ying, editors, *Frontiers in Algorithmics, 4th International Workshop, FAW 2010, Wuhan, China, August 11-13, 2010. Proceedings*, volume 6213 of *Lecture Notes in Computer Science*, pages 262–273. Springer, 2010. doi:10.1007/978-3-642-14553-7_25.

23 Jinfeng Dou, Thorsten Götte, Henning Hillebrandt, Christian Scheideler, and Julian Werthmann. Brief announcement: Distributed construction of near-optimal compact routing schemes for planar graphs. In Rotem Oshman, Alexandre Nolin, Magnús M. Halldórsson, and Alkida Balliu, editors, *Proceedings of the 2023 ACM Symposium on Principles of Distributed Computing, PODC 2023, Orlando, FL, USA, June 19-23, 2023*, pages 67–70. ACM, 2023. doi:10.1145/3583668.3594561.

24 Jinfeng Dou, Thorsten Götte, Henning Hillebrandt, Christian Scheideler, and Julian Werthmann. Distributed and parallel low-diameter decompositions for arbitrary and restricted graphs, 2024. arXiv:2411.19859.

25 Michael Elkin, Yuval Emek, Daniel A. Spielman, and Shang-Hua Teng. Lower-stretch spanning trees. *SIAM J. Comput.*, 38(2):608–628, 2008. doi:10.1137/050641661.

26 Michael Elkin, Arnold Filtser, and Ofer Neiman. Distributed construction of light networks. In *PODC '20: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, August 3-7, 2020*, pages 483–492. ACM, 2020. doi:10.1145/3382734.3405701.

27 Michael Elkin and Ofer Neiman. On efficient distributed construction of near optimal routing schemes: Extended abstract. In George Giakkoupis, editor, *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA, July 25-28, 2016*, pages 235–244. ACM, 2016. doi:10.1145/2933057.2933098.

28 Michael Elkin and Ofer Neiman. Near-optimal distributed routing with low memory. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, PODC '18, pages 207–216, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3212734.3212761.

29 Michael Elkin and Ofer Neiman. Distributed strong diameter network decomposition. *Theor. Comput. Sci.*, 922:150–157, 2022. doi:10.1016/J.TCS.2022.04.019.

30 Michael Elkin, Ofer Neiman, and Shay Solomon. Light spanners. *SIAM J. Discret. Math.*, 29(3):1312–1321, 2015. doi:10.1137/140979538.

31 Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.*, 69(3):485–497, 2004. doi:10.1016/j.jcss.2004.04.011.

32 Arnold Filtser. On Strong Diameter Padded Decompositions. In Dimitris Achlioptas and László A. Végh, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*, volume 145 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:21, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.APPROX-RANDOM.2019.6.

33 Arnold Filtser and Ofer Neiman. Light spanners for high dimensional norms via stochastic decompositions. *Algorithmica*, 84(10):2987–3007, 2022. doi:10.1007/S00453-022-00994-0.

34 Mohsen Ghaffari, Christoph Grunau, Bernhard Haeupler, Saeed Ilchi, and Václav Rozhon. Improved distributed network decomposition, hitting sets, and spanners, via derandomization. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 2532–2566. SIAM, 2023. doi:10.1137/1.9781611977554.CH97.

35 Mohsen Ghaffari and Bernhard Haeupler. Distributed algorithms for planar networks I: planar embedding. In George Giakkoupis, editor, *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA, July 25-28, 2016*, pages 29–38. ACM, 2016. doi:10.1145/2933057.2933109.

36 Mohsen Ghaffari and Bernhard Haeupler. Distributed algorithms for planar networks II: low-congestion shortcuts, mst, and min-cut. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 202–219. SIAM, 2016. doi:10.1137/1.9781611974331.ch16.

37 Mohsen Ghaffari and Bernhard Haeupler. Low-congestion shortcuts for graphs excluding dense minors. In Avery Miller, Keren Censor-Hillel, and Janne H. Korhonen, editors, *PODC '21: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, July 26-30, 2021*, pages 213–221. ACM, 2021. doi:10.1145/3465084.3467935.

38 Mohsen Ghaffari, Andreas Karrenbauer, Fabian Kuhn, Christoph Lenzen, and Boaz Patt-Shamir. Near-optimal distributed maximum flow. *SIAM J. Comput.*, 47(6):2078–2117, 2018. doi:10.1137/17M113277X.

39 Mohsen Ghaffari and Merav Parter. Near-optimal distributed DFS in planar graphs. In Andréa W. Richa, editor, *31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria*, volume 91 of *LIPIcs*, pages 21:1–21:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.DISC.2017.21.

40 Mohsen Ghaffari and Goran Zuzic. Universally-Optimal Distributed Exact Min-Cut. In *Proceedings of the 2022 ACM Symposium on Principles of Distributed Computing*, pages 281–291, Salerno Italy, July 2022. ACM. doi:10.1145/3519270.3538429.

41 Bernhard Haeupler, David Wajc, and Goran Zuzic. Universally-optimal distributed algorithms for known topologies. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1166–1179. ACM, 2021. doi:10.1145/3406325.3451081.

42 Taisuke Izumi, Naoki Kitamura, Takamasa Naruse, and Gregory Schwartzman. Fully polynomial-time distributed computation in low-treewidth graphs. In Kunal Agrawal and I-Ting Angelina Lee, editors, *SPAA '22: 34th ACM Symposium on Parallelism in Algorithms and Architectures, Philadelphia, PA, USA, July 11 - 14, 2022*, pages 11–22. ACM, 2022. doi:10.1145/3490148.3538590.

43 Fabian Kuhn and Philipp Schneider. Routing Schemes and Distance Oracles in the Hybrid Model. In Christian Scheideler, editor, *36th International Symposium on Distributed Computing (DISC 2022)*, volume 246 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 28:1–28:22, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.DISC.2022.28.

44 Christoph Lenzen and Boaz Patt-Shamir. Fast routing table construction using small messages: extended abstract. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 381–390, 2013. doi:10.1145/2488608.2488656.

45 Christoph Lenzen and Boaz Patt-Shamir. Fast partial distance estimation and applications. In Chryssis Georgiou and Paul G. Spirakis, editors, *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 - 23, 2015*, pages 153–162. ACM, 2015. doi:10.1145/2767386.2767398.

46 Christoph Lenzen, Boaz Patt-Shamir, and David Peleg. Distributed distance computation and routing with small messages. *Distributed Comput.*, 32(2):133–157, 2019. doi:10.1007/s00446-018-0326-6.

47 Reut Levi, Moti Medina, and Dana Ron. Property testing of planarity in the CONGEST model. *Distributed Comput.*, 34(1):15–32, 2021. doi:10.1007/S00446-020-00382-3.

48 Jason Li and Merav Parter. Planar diameter via metric compression. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 152–163. ACM, 2019. doi:10.1145/3313276.3316358.

49 Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Efficient Graph Minors Theory and Parameterized Algorithms for (Planar) Disjoint Paths*, pages 112–128. Springer International Publishing, Cham, 2020. doi:10.1007/978-3-030-42071-0_9.

50 Gary L. Miller, Richard Peng, Adrian Vladu, and Shen Chen Xu. Improved parallel algorithms for spanners and hopsets. In *Proc. of the 27th ACM symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 192–201, 2015. doi:10.1145/2755573.2755574.

51 Gary L. Miller, Richard Peng, and Shen Chen Xu. Parallel graph decompositions using random shifts. In Guy E. Blelloch and Berthold Vöcking, editors, *25th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA '13, Montreal, QC, Canada - July 23 - 25, 2013*, pages 196–203. ACM, 2013. doi:10.1145/2486159.2486180.

52 David Peleg. *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics, USA, 2000.

53 Neil Robertson and Paul D. Seymour. Graph minors. II. algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986. doi:10.1016/0196-6774(86)90023-4.

54 Neil Robertson and Paul D Seymour. Graph minors. xvi. excluding a non-planar graph. *Journal of Combinatorial Theory, Series B*, 89(1):43–76, 2003. doi:10.1016/S0095-8956(03)00042-X.

55 Václav Rozhon, Michael Elkin, Christoph Grunau, and Bernhard Haeupler. Deterministic low-diameter decompositions for weighted graphs and distributed and parallel applications. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 1114–1121. IEEE, 2022. doi:10.1109/FOCS54457.2022.00107.

56 Václav Rozhon, Christoph Grunau, Bernhard Haeupler, Goran Zuzic, and Jason Li. Undirected (1+ϵ)-shortest paths via minor-aggregates: near-optimal deterministic parallel and distributed algorithms. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 478–487. ACM, 2022. doi:10.1145/3519935.3520074.

57 Philipp Schneider. *Power and limitations of hybrid communication networks*. PhD thesis, University of Freiburg, Freiburg im Breisgau, Germany, 2023. URL: https://freidok.uni-freiburg.de/data/232804.

58 Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *J. ACM*, 51(6):993–1024, November 2004. doi:10.1145/1039488.1039493.

Data-Driven Solution Portfolios

Marina Drygala
EPFL, Lausanne, Switzerland

Silvio Lattanzi
Google Research, Barecelona, Spain

Andreas Maggiori
Columbia University, New York, NY, USA

Miltiadis Stouras[1]
EPFL, Lausanne, Switzerland

Ola Svensson[1]
EPFL, Lausanne, Switzerland

Sergei Vassilvitskii
Google Research, New York, NY, USA

Abstract

In this paper, we consider a new problem of portfolio optimization using stochastic information. In a setting where there is some uncertainty, we ask how to best select k potential solutions, with the goal of optimizing the value of the best solution. More formally, given a combinatorial problem Π, a set of value functions $\mathcal{V}$ over the solutions of Π, and a distribution $\mathcal{D}$ over $\mathcal{V}$, our goal is to select k solutions of Π that maximize or minimize the expected value of the *best* of those solutions. For a simple example, consider the classic knapsack problem: given a universe of elements each with unit weight and a positive value, the task is to select r elements maximizing the total value. Now suppose that each element's weight comes from a (known) distribution. How should we select k different solutions so that one of them is likely to yield a high value?

In this work, we tackle this basic problem, and generalize it to the setting where the underlying set system forms a matroid. On the technical side, it is clear that the candidate solutions we select must be diverse and anti-correlated; however, it is not clear how to do so efficiently. Our main result is a polynomial-time algorithm that constructs a portfolio within a constant factor of the optimal.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases solution portfolios, data-driven algorithm design, matroids

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.46

Related Version *Full Version*: https://arxiv.org/abs/2412.00717

1 Introduction

Worst-case analysis has long been the prevailing standard for assessing an algorithm's performance. However, this approach often falls short in capturing the real-world performance of algorithms, as instances encountered in practice often differ significantly from those that define worst-case scenarios. To address this discrepancy, the field of *Data-Driven Algorithm Design* has sought to create algorithms that utilize past data about a problem, either implicitly or explicitly, and provably achieve superior performance on the typical instances that arise in practical applications. For surveys of this area see [12, 1].

[1] Supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number MB22.00054.



© Marina Drygala, Silvio Lattanzi, Andreas Maggiori, Miltiadis Stouras, Ola Svensson and Sergei Vassilvitskii;
licensed under Creative Commons License CC-BY 4.0



Building on this line of research, we study the portfolio optimization problem as a simple framework for speeding up algorithms using historical data; a method with potential applications across various combinatorial problems. Specifically, given a combinatorial problem Π, a solution set $\mathcal{I}$, and a distribution $\mathcal{D}$ over value functions (which map solutions to values, with each function representing a different scenario), our objective is to compute a portfolio of k solutions that maximizes (or minimizes) the expected value (or cost) of the best among the k solutions. Starting from a different perspective, Kleinberg, Papadimitriou and Raghavan [9] have defined a very similar family of optimization problems, called "Segmentation Problems". The Portfolio Optimization problem can be seen as a stochastic version of the aforementioned family of problems; actually the two definitions are effectively equivalent when the distribution over value functions has a polynomial-sized support (for more details see Related Work, Section 1.2).

The Portfolio Optimization problem captures many natural questions. The problem mentioned in the abstract–how to select k bundles of r items each so that the expected value of the best bundle is maximized–is only one of them. Another practical application is finding the shortest path between two points (e.g., home and work) in a city, under varying daily traffic conditions. Here, the solution set consists of all possible source-destination paths and the value functions assign weights to each path based on specific traffic scenarios. The distribution over value functions models the stochastic nature of traffic. Rather than rerunning a shortest-path algorithm for each new instance, one could leverage the statistical knowledge about traffic patterns to precompute a few paths that cover different likely scenarios (e.g., peak morning traffic, light nighttime traffic, etc.). This approach, then, allows us to quickly evaluate the precomputed paths under new traffic conditions and select the best option without the need to resolve the problem from scratch each time.

Interestingly, this stochastic formulation also captures problems outside the area of speeding up algorithms. In particular, our objective captures any stochastic problem where one needs to select a set of dependent objects with the goal of maximizing the expectation of their maximum. A practical example of this is sports betting pools, which was recently explored in [6]. The authors examine the scenario of participating in a betting pool for a basketball tournament, where individuals can pay a fee to submit a prediction for the outcome of all tournament matches, with the potential to win a substantial monetary prize if their predictions are accurate. Given a budget constraint on the number of entries that a person can submit, the technical problem essentially reduces to computing a portfolio of entries that maximizes the probability that one of the selected entries succeeds.

From a technical standpoint, in the maximization variant of our problem, the value of a portfolio is a submodular function. Therefore, one can find an approximately optimal portfolio by running the *Greedy* algorithm, with a running time that is polynomial in the size of the solution set $\mathcal{I}$. However, for most interesting problems, e.g., spanning trees, entries in betting pools, etc., the solution set is given implicitly and is usually exponentially large in the size of the input, making *Greedy* inefficient. This raises the natural question of whether one can compute approximately optimal portfolios, for interesting problems, with a running time that is polynomial in the size of the input.

1.1 Our contributions

The first important contribution of our paper is conceptual: we formulate the Portfolio Optimization problem from a new, stochastic, viewpoint and show that it captures many interesting scenarios.

On the technical side, we focus on constructing solution portfolios for the fundamental problem of optimization under matroid contraints. We examine the simplified case where each element of the matroid's ground set takes a 0-1 value, independently, with some known probability, and the value of a set is the sum of the values of its elements. Despite its simplicity, this case captures many core challenges of constructing solution portfolios and proves to be technically challenging. As we discuss in Section 3, this problem differs fundamentally from classical problems in the area of randomized algorithms, because constructing effective portfolios requires leveraging the "anti-concentration" properties of various solutions. Additionally, employing standard tools from the literature, such as contention resolution schemes, presents challenges due to the nature of our objective function. In particular, the value of a portfolio depends heavily on "abnormal" events, such as significant deviations from the expected value, occurring in one of its k solutions. However, contention resolution schemes do not offer per-instance guarantees but rather only work on expectation. Therefore, we need to take extra care to apply these results while conditioning on those events.

The main technical contribution of our work is to design an algorithm with polynomial running time in the size of the matroid's ground set which constructs a portfolio that is a $\Theta(1)$-approximation of the optimal portfolio.

We give a high-level description of our techniques in Section 3 and present a simpler algorithm for the case of uniform matroids in Section 4. In the full version of the paper we show how this algorithm can be extended to work for all matroids.

1.2 Related work

The related work can be broadly categorized into three main directions. The first is the study of Segmentation Problems, initially introduced by Kleinberg, Papadimitriou and Raghavan [9]. This work, inspired by data mining techniques for market segmentation, defines a new class of optimization problems named *Segmentation Problems*. For any combinatorial optimization problem and a set S of different cost vectors for this problem, the corresponding segmentation problem asks to partition the set S into several *segments* and pick a separate solution for each segment, so that the total cost is minimized. The Portfolio Optimization problem can be seen as a stochastic variant of the aforementioned family of problems, where instead of a fixed set of cost vectors, one has access to a distribution over cost vectors and aims to optimize the expected value of the constructed portfolio. In fact, the two formulations are effectively equivalent when the distribution over cost vectors has a polynomial-sized support. This new, stochastic, viewpoint can accommodate a wider variety of applications where randomness is inherent in the problem at hand (e.g., sports betting, as introduced earlier). Besides that, it also enables the formulation of elegant and technically challenging theoretical questions, like the case where every element of a groundset takes a value independently of the other elements. Another adaptation of the family of "Segmentation Problems" is due to Gupta, Moondra and Singh [8] who examined a "robust" version of the problem. In this variant, the objective is to identify a small set of solutions that guarantees a good approximation for each one of the cost vectors of interest. This approach was motivated by fairness concerns, aiming to guarantee a good approximation across various cost vectors that may arise due to different fairness constraints.

On the technical side, Kleinberg et al. [9] study the Catalogue Segmentation problem, i.e. constructing portfolios for linear maximization with cardinality constraints. For the case where elements take values in $\{0, 1\}$, they design an algorithm that runs in time $O(n^{k \log k}/\delta)$ and produces a portfolio that is a $(1 - \delta)$-approximation, under the assumption that on every instance a large fraction of the items have value 1. Our result is a $O(1)$ approximation

with a running time that is polynomial in both n and k, without the need of a density assumption, but for the case where the value of each element is independent of the values of other elements. Furthermore, our results extend to more general settings, accommodating any matroid constraints on the ground set.

The second related line of work is the area of Data-Driven Algorithm Design and specifically speeding up an algorithms' execution by utilizing data. Some of these attempts include speeding up: the *Greedy* algorithm for submodular maximization [2, 15], the Hungarian algorithm for calculating maximum matchings [7], primal-dual algorithms for various graph problems [3], algorithms for flow problems [4, 5] and the Bellman-Ford algorithm [11]. The key distinction of our work lies in defining a new set of problems and providing general techniques for solving them.

A third relevant area of research is stochastic probing and online decision-making (for a detailed survey, see [14]). These problems typically involve a ground set of elements, each with an unknown stochastic weight sampled from a known distribution. The algorithm can probe certain elements – often incurring a cost – in order to reveal their weights. Based on these observations, it then selects a feasible subset and is rewarded with the weight of the selected items. A key distinction between this area and the Portfolio Optimization problem is that, in our setting, the solutions are chosen entirely offline, without observing any weights, and remain fixed across all possible realizations.

Our problem is more closely aligned with a non-adaptive strategy for stochastic probing. However, the variants of non-adaptive strategies studied in the literature typically impose only partial restrictions on the algorithm's flexibility – for instance, requiring the algorithm to preselect which elements to probe but still allowing it to form its solution after observing the probed weights. In contrast, in the Portfolio Optimization problem, the selection is entirely offline, with no adjustments allowed after weights are realized. For this reasons, the benchmarks of the two problems are also different: non-adaptive strategies are evaluated against the optimal value achievable in hindsight, whereas our benchmark is the best offline strategy with the same constraints as the algorithm.

2 Problem statement and preliminaries

2.1 Portfolio Optimization

In this section we define the Portfolio Optimization problem in its general form, for any classical combinatorial problem. In the next section, we define the Matroid Portfolio Optimization problem, which is the portfolio problem for a special case of maximization with matroid constraints.

We present the maximization variant of our problem, with the minimization version defined in a similar manner. For any combinatorial problem Π, the Portfolio Optimization problem is described by a tuple $\mathcal{J} = (\mathcal{I}, \mathcal{V}, \mathcal{D}, k)$ where $\mathcal{I}$ is the set of feasible solutions of problem Π, $\mathcal{V}$ is a set of value functions from $\mathcal{I}$ to $\mathbb{R}_{\geq 0}$, $\mathcal{D}$ is a distribution over $\mathcal{V}$ and k is a natural number that describes the desired size of the portfolio.

Our goal is to select k solutions from $\mathcal{I}$ so as to maximize the expected value of the *best* of those solutions. That is, for a collection of k sets $\mathcal{S} = \{S_1, \ldots, S_k\}$ such that $S_i \in \mathcal{I}, \forall i \in \{1, \ldots, k\}$ we define its value as:

$$\text{value}(\mathcal{S}) = \mathop{\mathbf{E}}_{v \sim \mathcal{D}} \left[\max_{S_i \in \mathcal{S}} v(S_i) \right].$$

Formally, we want to solve the following optimization problem.

$$\begin{aligned} &\text{maximize value}(\mathcal{S}) \\ &\quad \text{s.t.} \quad \mathcal{S} = \{S_1, ..., S_k\} \text{ and } S_i \in \mathcal{I}, \forall i \in [k]. \end{aligned}$$

We remark that the solutions $S_1, \ldots, S_k$ are chosen *offline*, without observing the realized value function, but rather only by using our knowledge about the distribution $\mathcal{D}$.

For any input tuple $\mathcal{J}$, we denote by $\mathcal{O}(\mathcal{J}) = \{O_1(\mathcal{J}), \ldots, O_k(\mathcal{J})\}$ the optimal portfolio, that is the maximizer of the above optimization problem. To ease notation, whenever the input and the algorithm are clear from the context we use $\mathcal{O} = \{O_1, \ldots, O_k\}$ and OPT to denote the optimum solution and its value respectively.

For an algorithm ALG we denote by $\mathcal{S}_{\text{ALG}(\mathcal{J})}$ its output on input $\mathcal{J}$. We say that ALG is a c-approximation if it always outputs a collection of feasible solutions and:

$$\mathop{\mathbf{E}}_{\text{ALG}} \left[\text{value}\left(\mathcal{S}_{\text{ALG}(\mathcal{J})}\right)\right] \geq c \cdot \text{value}\left(\mathcal{O}(\mathcal{J})\right), \ \forall \mathcal{J}$$

where the expectation is taken over the internal randomness of ALG.

In the maximization variant of the problem, the value of a set of solutions, defined as the expected maximum of their values, is a monotone submodular function, as demonstrated by Kleinberg and Raghu [10]. Consequently, the celebrated *Greedy* algorithm of [13] is a $(1-1/e)$ approximation that runs in polynomial time in the size of the solution set $\mathcal{I}$. Without imposing any further restrictions to the problem, this is the best approximation ratio one can achieve, as the Max-k-Cover problem can be reduced to the Portfolio Optimization problem. The details of the reduction are deferred to the full version of the paper.

Although Greedy achieves the optimal approximation ratio for this problem, its running time is impractical for most interesting applications like max-k-cover, knapsack, shortest paths, spanning trees, etc. In all of the aforementioned applications, the set of feasible solutions is given implicitly and is exponentially large in the size of the input. Therefore, one needs to have a running time that is polynomial in the description of the solution set, rather than its size. This raises the natural question of whether one can compute approximately optimal portfolios, for interesting problems, with a running time that is polynomial in the description of the solution set $\mathcal{I}$.

In this work, we focus on the fundamental case of optimization over matroid contraints and answer the latter question affirmatively for a natural distribution over value functions. We formulate this problem in the following section.

2.2 Matroid Portfolio Optimization

The Matroid Portfolio Optimization problem is a special case of the Portfolio Optimization problem, where the set of feasible solutions, $\mathcal{I}$, is the family of independent sets of an underlying matroid $M = (E, \mathcal{I})$ (see Definition 1). The set $\mathcal{V}$ of value functions is the set of all *additive* set functions that map the elements of E to $\{0, 1\}$. Formally, for any value function $u \in \mathcal{V}$, element $e \in E$ and subset $S \subseteq E$ we have: $u(\{e\}) \in \{0, 1\}$ and $u(S) = \sum_{e' \in S} u(\{e'\})$. Throughout the paper we use the term "active" to denote an element $e \in E$ that takes value one, under a specific value function, and the term "inactive" to describe an element with value zero.

We assume that $\mathcal{D}$ is such that each element $e \in E$ is active with probability p_e, independently of the values of the rest of the elements of E. Formally,

$$\forall S \subseteq E : \mathop{\mathbf{Pr}}_{u\sim\mathcal{D}}\left[\bigwedge_{e\in S}\{u(\{e\}) = 1\}\right] = \prod_{e\in S} p_e$$

To simplify the notation we define an equivalent distribution $\mathcal{D}'$ over subsets of E, that represent the set of active elements, such that

$$\forall e \in E : \mathop{\mathbf{Pr}}_{u\sim\mathcal{D}}[u(\{e\}) = 1] = \mathop{\mathbf{Pr}}_{A\sim\mathcal{D}'}[e \in A]\,.$$

and

$$\forall S \subseteq E : \mathop{\mathbf{Pr}}_{A\sim\mathcal{D}'}\left[\bigwedge_{e\in S}\{e \in A\}\right] = \prod_{e\in S} p_e$$

In other words, instead of sampling a value function that assigns $\{0, 1\}$ values to the elements, we can equivalently sample the set of active elements and, thus, switch to distributions over subsets of E. Also, the value of any set $S \subseteq E$ will now be equal to the random variable $|S \cap A|$, where A is the random set denoting the active elements.

For the remainder of the paper, the input to the Matroid Portfolio Optimization problem will be described by a triplet $\mathcal{J} = (M, k, \mathcal{D})$, where $M = (E, \mathcal{I})$ is a matroid over the ground set E, k a natural number describing the desired size of the portfolio and $\mathcal{D}$ is a distribution over subsets of E such that each element $e \in E$ is included in a sample, independently, with probability p_e. For a collection $\mathcal{S} = \{S_1, \ldots, S_k\}$ such that $S_i \in \mathcal{I}, \forall i \in \{1, \ldots, k\}$, we define its value as

$$\text{value}(\mathcal{S}) = \mathop{\mathbf{E}}_{A\sim\mathcal{D}}\left[\max_{S_i\in\mathcal{S}} |S_i \cap A|\right].$$

2.3 Preliminaries

In this subsection we introduce notation and give some preliminary definitions. First, for any positive integer ℓ we use the shorthand $[\ell]$ to denote the set $\{1, \ldots, \ell\}$. In addition, we give the definition of matroids below.

▶ **Definition 1.** *A pair $M = (E, \mathcal{I})$ is called a matroid if E is a finite ground set and $\mathcal{I}$ is a non-empty collection of subsets of E such that*

1. *If $I \in \mathcal{I}$ and $J \subseteq I$, then $J \in \mathcal{I}$,*
2. *If $I, J \in \mathcal{I}$ and $|I| < |J|$, then $I + e \in \mathcal{I}$ for some $e \in J \setminus I$.*

For a matroid $M = (E, \mathcal{I})$ the elements of $\mathcal{I}$ are called the independent sets of M. The rank function of a matroid M, $r_M : 2^E \to \mathbb{N}$, maps each set $U \subseteq E$ to the size of the largest independent set contained in U. We use the term "rank of a matroid M" to denote the rank r of the ground set, i.e. $r = r_M(E)$. A set $B \subseteq E$ is called a base if and only if it's a maximum size independent set. In addition, the span of a set $S \subseteq E$ is defined as $\text{span}_M(S) = \{e \in E : \text{rank}(S \cup \{e\}) = \text{rank}(S)\}$.

The matroid polytope $\mathcal{P}(M)$, of a matroid M, is a subset of $\mathbb{R}^{|E|}$ that is defined by the following sets of inequalities, where r_M is the rank function of M.

$$\mathcal{P}(M) = \left\{ \begin{array}{ll} \displaystyle\sum_{e\in U} x_e \le r_M(U), & \forall U \subseteq E \\ x_e \ge 0, & \forall e \in E \end{array} \right\}$$

It is well known that one can check whether some $x \in \mathbb{R}^{|E|}$ lies in the matroid polytope, that is $x \in \mathcal{P}(M)$, in polynomial time in the size of E.

3 Overview of our Techniques

The core difficulty of the Matroid Portfolio Optimization problem stems from its objective function. Indeed, calculating or bounding, the expectation of the maximum of k random variables is a difficult task, especially when the random variables are dependent. Apart from that, optimizing under this objective requires to look at the problem from a perspective that differs with what we are used to in the analysis of randomized algorithms. Usually, we argue that our solutions have a good enough expectation and that they reach this expectation with a reasonable probability. However, in order to construct a good portfolio we are interested in the "anti-concentration" properties of the solutions that we pick, meaning that we want each solution to be able to greatly surpass its expected value with a reasonable probability. Intuitively, if the values of our solutions were i.i.d. random variables, we would want them to be somewhat "heavy-tailed" so that after k independent samples there would be a good probability that we observe a high outlier. On the other hand, if the solutions that we pick were highly concentrated around their expected value, then we would observe almost no benefit by taking the maximum of several random variables.

In this section, we first highlight our ideas for the simpler case of Matroid Portfolio Optimization for uniform matroids and then explain how they can properly be generalized to work for all matroids. As a reminder, in the Matroid Portfolio Optimization problem on uniform matroids, we are given a ground set of n elements and we want to construct a portfolio $\mathcal{P}$ consisting of k subsets of the elements, each of size r. The value of the i-th element is an independent Bernoulli random variable with probability p_i and we are interested in maximizing the value of the portfolio that we construct.

Even in this simple case, understanding the core trade-off of using a higher probability element multiple times across solutions versus replacing it with independent elements of lower probability is a challenging task. In order to build some intuition, in the next subsection, we discuss some natural approaches for this simple case and show why they fail to produce a portfolio that is a constant-factor approximation of the optimal portfolio.

3.1 Natural approaches that fail

The first approach for this problem is simply to pick the k independent sets with the highest expected value. Of course, it makes sense to enforce that these independent sets are *disjoint*, since we want them to "complement" each other. If these sets had large pairwise intersections, we would observe no benefit from selecting k of them instead of one. In other words, a natural strategy is to pick the highest expectation subset, remove it from the ground set, continue by picking the highest expectation subset of the remaining elements and so on. However, this approach will fail to construct a good portfolio.

Intuitively, in some cases it is better to restrict ourselves to considerably less disjoint subsets and complete our portfolio by combining those subsets in a clever way, instead of using new subsets that might have lower expectation. For example, consider the instance where we need to pick k subsets of size $r = k$ and we have $n = r^2$ elements available. Let the first $k \log k$ elements have activation probability $1/k$, and the rest to have activation probability $1/k^2$. The approach described above will form k disjoint subsets, out of which $\log k$ will behave as independent binomials with expectation 1 and the rest will be independent binomials with expectation $1/k$. The value of this portfolio will be dominated by the expectation of the maximum of the first $\log k$ binomials, which is $O(\log \log k)$. Surprisingly, instead of using the lower probability elements, one can pick k solutions by uniformly combining parts of the first $\log k$ subsets and construct a portfolio that has value $\Theta(\log k / \log \log k)$, which is

asymptotically optimal (we describe this construction in the full version of the paper). This example showcases the power of having a clever "mixing" strategy as this can boost the portfolio to achieve an exponentially better value than a portfolio that only uses disjoint solutions. In fact, in this example, a strategy that picks only disjoint solutions would need k disjoint subsets of expectation 1 in order to achieve the same value as the one achieved by uniformly mixing $\log k$ subsets of expectation 1.

3.2 Main ideas

Filtering out elements

From the previous example, it becomes evident that a candidate algorithm should have a filtering procedure that discards some elements of the ground set, and a mixing strategy, which combines the remaining elements to form the desired subsets. It is natural to wonder if one can commit to the simplest possible mixing strategy (i.e. sampling elements uniformly) and try to find a filtering rule that would make this strategy work. Note that the nearly-optimal solution constructed for the previous example actually fits into this framework.

In the simpler case of uniform matroids, designing the filtering procedure can be reduced to first sorting the elements in decreasing order of their probabilities and then selecting a prefix of this order. A key observation in our work is that there always exists a prefix of the elements that admits a good portfolio under the simplest mixing strategy, i.e. forming subsets by uniform sampling. This observation directly gives us a polynomial-time constant-approximation algorithm, as one can try all possible n prefixes, generate the corresponding portfolios through uniform sampling and keep the best one of them after estimating their values. We formally present this algorithm in Section 4 and build up to our algorithm for general matroids in the full version of our paper.

Analyzing the value of the produced portfolio

The simplicity of our mixing strategy allows us to bypass dealing with the dependencies of the solutions we form when we want to lower bound the expectation of their maximum. On a high level, to analyze the value of our portfolio, we first fix the randomness of the instance, by conditioning on an outcome for the active elements, and then analyze the expected value of a sampled solution over the internal randomness of the algorithm. Once we have fixed the outcome of the active elements, the *values* of the sampled solutions are simply binomial random variables that only depend on the internal randomness of the sampling procedure. Critically, this makes the values of the sampled solutions *independent* random variables which makes them much easier to work with. On the other hand, if we had first fixed the portfolio produced by the algorithm and then tried to analyze its value over the randomness of the instance, we would have to analyze the dependencies of the picked solutions and how these influence the value of the portfolio, which is a much harder task. This analysis technique, of course, comes with its own challenges as selecting the appropriate event to condition on is not a trivial task.

Generalizing to all matroids

The filtering procedure described above fails to work beyond uniform matroids, simply because it ignores the underlying matroid structure of the elements. For example, consider the case of the graphic matroid of a graph G that consists of a clique on $\sqrt{n}$ vertices and a simple path of $n - \sqrt{n}$ vertices. Let the activation probabilities of the edges of the clique

to be $1/\sqrt{n}$ and the activation probabilities of the edges of the path to be $1/\sqrt{n} - \epsilon$ for some small $\epsilon > 0$. Ordering the elements by their activation probabilities and selecting a prefix of this ordering goes towards the wrong direction as one should prioritize taking edges from the path over taking edges from the clique. Indeed, taking the path as our only solution would give us an expected value of $\Theta(\sqrt{n})$, whereas any portfolio of $k = O(n)$ spanning trees of the clique has value at most $O(\log n / \log \log n)$.

In order to overcome this issue, we change our filtering procedure to select a certain number of disjoint, high-expectation, independent sets of the matroid. In other words, we create an ordering of disjoint independent sets of decreasing expectation and we take a prefix of this ordering. Constructing this ordering can simply be achieved by finding the biggest expectation base of the matroid through the Greedy algorithm, then restricting the matroid to the remaining elements and repeating. As in the case of uniform matroids, we prove that there always exists a prefix of this ordering that admits a nearly-optimal portfolio after sampling elements uniformly and then passing them through a contention resolution scheme.

Finally, the biggest technical challenge of this problem is analyzing the value of the produced portfolios in the case of general matroids. To be more precise, the value of a portfolio heavily relies on "abnormal" events happening in one of its k solutions. For example, if we fix a set of active elements, and analyze the expected value of the sampled solutions over the internal randomness of the algorithm, the expected value of each solution will be the number of active elements we expect to sample in one trial. Crucially, one of the k trials will sample much more active elements than its expectation, and this trial will be responsible for the value of the portfolio. At this point, this solution will be passed through a contention resolution scheme to be trimmed down to an independent set. However, contention resolution schemes do not have per-instance guarantees but only work in expectation. Therefore, there is no guarantee that, when this abnormal event happens, the active elements are not going to be discarded by the contention resolution scheme. We overcome this issue by conditioning on appropriate events that do not entirely fix the randomness of either the instance or the algorithm, but still allow us to use the guarantees of a contention resolution scheme and argue that one of the sampled solutions reaches a near-optimal value. This is the main result of our work which is formally stated below.

▶ **Theorem 2.** *If $\mathcal{D}$ is a product distribution then there exists an algorithm that is achieves $\Theta(1)$-approximation for the k-portfolio solution problem and has a polynomial time complexity.*

4 Warm-up: An $O(1)$ approximation for Uniform Matroids

In this section we formally introduce some of the ideas described above by presenting our algorithm for the case of uniform matroids. We remind the reader that the input is described by a triplet $\mathcal{J} = (M, k, \mathcal{D})$ where $M = (E, \mathcal{I})$ is a (uniform in this section) matroid with rank r, k is the size of our portfolio and $\mathcal{D}$ is a product distribution where each element $e \in E$ is active with probability p_e. For simplicity, we order elements in E in decreasing order of their probabilities. We denote by e_i the i-th element in this order and by p_i its activation probability. In addition, throughout this section we assume that $\text{OPT} \geq 200$. In the full version of the paper, we prove that lower bounding OPT by a large constant is without loss of generality.

As mentioned in Section 3, our algorithm filters out some elements of the ground set and then produces the k solutions of its portfolio by sampling elements uniformly at random. We prove that for any instance of the problem, there exists a prefix of the elements (when ordered by decreasing activation probability) that admits a near-optimal portfolio under

uniform sampling. Therefore, the algorithm simply needs to try all n prefixes, estimate the values of the constructed portfolios and output the best one of them. We give the pseudocode of this strategy in Algorithm 1.

Algorithm 1 An algorithm for Uniform Matroids.

```
function CREATE-PORTFOLIO(J = (M, k, D))          ▷ M: uniform matroid of rank r
    Portfolios ← []
    for i = 1, ..., n do
        Prefix ← {1, ..., i}                      ▷ Elements are ordered with decreasing p_i
        Portfolios[i] ← PORTFOLIO-FROM-PREFIX(Prefix, k, r)
    end for
    Estimate the values of the n portfolios
    Return the portfolio with the biggest estimated value
end function

function PORTFOLIO-FROM-PREFIX(Prefix, k, r)
    P ← {}
    for i = 1, ..., k do
        Let V_i = {s_1, ... s_r} be r uniformly random samples from Prefix
        S ← non-duplicate elements of V_i
        P ← P ∪ S
    end for
    return P
end function
```

In order to prove that Algorithm 1 is an $O(1)$-approximation, it suffices to show that there exists a prefix on which sampling elements uniformly creates a portfolio with value at least $\Theta(1) \cdot \mathrm{OPT}$. Indeed, the value of each solution can be estimated using standard techniques within a small factor with polynomially many samples with high probability. Since there is only n solutions the estimate is close for all of them with good probability (by union bound) and we output the best solution. The prefix on which we will focus, is the largest prefix whose expected value is at most $\mathrm{OPT}/2$. More formally, let M be the largest index such that

$$\sum_{i=1}^{M} p_i < \mathrm{OPT}/2.$$

By the definition of M, we know that $\sum_{i=1}^{M+1} p_i \geq \mathrm{OPT}/2$, therefore we also get that

$$\sum_{i=1}^{M} p_i \geq \mathrm{OPT}/2 - p_{M+1} \geq \mathrm{OPT}/2 - 1 \geq \mathrm{OPT}/3,$$

where in the last inequality we used that $\mathrm{OPT} \geq 6$.

Our first observation is that there exists a portfolio which only uses elements outside of the selected prefix, i.e. elements with lower activation probability than what our algorithm has picked, and that achieves value at least $\mathrm{OPT}/2$. This portfolio is the restriction of the optimal one to the elements outside of the prefix. Let $H = \{e_1, \ldots, e_M\}$ and $L = \{e_{M+1}, \ldots, e_n\}$. The aforementioned claim is stated formally in the following lemma:

▶ **Lemma 3.** $\mathbf{E}_{A \sim \mathcal{D}} \left[\max_{O_i \in \mathcal{O}} |O_i \cap L \cap A|\right] \geq \frac{\mathrm{OPT}}{2}$.

Proof of Lemma 3.

$$\begin{aligned}
\mathrm{OPT} &= \mathop{\mathbf{E}}_{A\sim\mathcal{D}}\left[\max_{O_i\in\mathcal{O}} |O_i \cap A|\right] && (1)\\
&= \mathop{\mathbf{E}}_{A\sim\mathcal{D}}\left[\max_{O_i\in\mathcal{O}} |O_i \cap H \cap A| + |O_i \cap L \cap A|\right] && (2)\\
&\leq \mathop{\mathbf{E}}_{A\sim\mathcal{D}}\left[|A \cap H| + \max_{O_i\in\mathcal{O}} |O_i \cap L \cap A|\right] && (3)\\
&\leq \frac{\mathrm{OPT}}{2} + \mathop{\mathbf{E}}_{A\sim\mathcal{D}}\left[\max_{O_i\in\mathcal{O}} |O_i \cap L \cap A|\right] && (4)\\
\Rightarrow \mathop{\mathbf{E}}_{A\sim\mathcal{D}}&\left[\max_{O_i\in\mathcal{O}} |O_i \cap L \cap A|\right] \geq \frac{\mathrm{OPT}}{2}, && (5)
\end{aligned}$$

where from (2) to (3) we used that $O_i \cap H \cap A \subseteq H \cap A$ for all i and from (3) to (4) that $\mathbf{E}_{A\sim\mathcal{D}}[|A \cap H|] = \sum_{i=1}^{M} p_i \leq \frac{\mathrm{OPT}}{2}$. ◀

The next observation we make is that if someone had access to $k \cdot r$ independent copies of the element e_{M+1}, then by using these elements they could construct a portfolio that has value at least OPT/2.

▶ **Lemma 4.** *Let $B_1, \ldots, B_k$ be k i.i.d random variables following $Bin(r, p_{M+1})$. Then,*

$$\mathbf{E}\left[\max_{i\in[k]} B_i\right] \geq \frac{\mathrm{OPT}}{2}.$$

Proof of Lemma 4. From Lemma 3, we know that there exist k dependent Poisson Binomials, namely the solutions picked by the optimal portfolio restricted to L, each of which has at most r trials, trial probabilities at most p_{M+1}, and whose expected maximum is at least OPT/2. The proof follows from the fact that one construct the desired Binomials, $B_1, \ldots, B_k$, by starting from the Poisson Binomials and doing the following transformations:

1. Increase the number of trials of all Poisson Binomials to r by augmenting new independent Bernoulli random variables each having probability p_{M+1}.
2. Make the Poisson Binomials independent by introducing new independent copies for the elements that are shared across many solutions.
3. Transform the Poisson Binomials into Binomials by increasing all probabilities to p_{M+1}.

All of the above transformations do not decrease the expectation of the maximum of the random variables, since (1) the augmented variables stochastically dominate the previous ones, (2) making the random variables independent can only increase the expectation of their maximum, because solutions with shared elements are positively correlated and "fail" together and (3) increasing the probabilities of the trials can also only increase the expectation of their maximum. ◀

The previous lemma gives us a "target" for analyzing the expected value of our portfolio. After conditioning on an appropriate constant probability event for the activation of elements, we argue that the *values* of the sampled solutions are independent Binomial random variables with trial probability close to p_{M+1}. We define such event as H having at least OPT/12 active elements and prove that the latter happens with probability at least 1/2. To that end, let $W = \left\{\tilde{E} \subseteq E : |\tilde{E} \cap H| \geq \mathrm{OPT}/12\right\}$.

▶ **Lemma 5.** $\mathbf{Pr}_{A\sim\mathcal{D}}[A \in W] \geq 1/2$.

Proof of Lemma 5. The left hand side of the desired inequality can be written as:

$$\begin{aligned}
\Pr_{A\sim\mathcal{D}}[A \in W] &= \Pr_{A\sim\mathcal{D}}\left[|A\cap H| \geq \frac{\mathrm{OPT}}{12}\right] && (1)\\
&\geq \Pr_{A\sim\mathcal{D}}\left[|A\cap H| \geq \frac{\mathbf{E}[|A\cap H|]}{4}\right] && (2)\\
&\geq \Pr_{A\sim\mathcal{D}}\left[||A\cap H| - \mathbf{E}[|A\cap H|]| \leq \frac{3}{4}\mathbf{E}[|A\cap H|]\right] && (3)\\
&\geq 1 - 2e^{-\left(\frac{3}{4}\right)^2\cdot\frac{1}{3}\cdot\mathbf{E}[|A\cap H|]} && (4)\\
&\geq \frac{1}{2} && (5)
\end{aligned}$$

where for (1) and (4) we used that $\mathbf{E}_{A\sim\mathcal{D}}[|A\cap H|] = \sum_{i=1}^{M} p_i \geq \frac{\mathrm{OPT}}{3} \geq 10$ and for (3) we used a Chernoff Bound for the Binomial random variable $|A\cap H|$. ◀

We continue by proving Lemma 6 for the expected value of the sampled multisets V_i. For simplicity, we slightly abuse notation and for any set $\tilde{E} \subseteq E$ we use $|V_i \cap \tilde{E}|$ to denote the sum $\sum_{x\in V_i} \mathbb{1}\{x \in \tilde{E}\}$.

▶ **Lemma 6.** *For any $\tilde{E} \in W$, it holds that*

$$\mathop{\mathbf{E}}_{\mathrm{ALG}}\left[\max_{i\in[k]} |V_i \cap \tilde{E}|\right] \geq \mathrm{OPT}/24.$$

Proof of Lemma 6. When we have fixed an outcome $\tilde{E} \in W$ for the active elements, the values of the sampled multisets, V_i, depend only on the internal randomness of our sampling procedure and are, thus, independent random variables. In addition, the algorithm samples elements uniformly at random from H. Therefore the probability that a sampled element is active is

$$\frac{|\tilde{E}\cap H|}{|H|} \geq \frac{\mathrm{OPT}}{12\cdot|H|} \geq \frac{\sum_{i\in H} p_i}{12\cdot|H|} \geq \frac{p_{M+1}}{12},$$

where for the first inequality we used that the definition of W, for the second inequality we used that $\sum_{i\in H} p_i \leq \mathrm{OPT}/2$ from the definition of the prefix H, and for the third inequality we used that $\forall i \in H : p_i \geq p_{M+1}$.

Therefore, the random variables $|V_i \cap \tilde{E}|$ are independent Binomials with trial probability at least $p_{M+1}/12$ for all i. Intuitively, this means that the expectation of their maximum should be close to the expectation of the maximum of k independent binomials with trial probability p_{M+1}, which by Lemma 4 is at least $\Theta(1)\cdot\mathrm{OPT}$.

More formally, let $B_1, \ldots, B_k$ be iid random variables following $Bin(r, p_{M+1})$ and $B'_1, \ldots, B'_k$ be iid random variables following $Bin(r, p_{M+1}/12)$. Then for any $\tilde{E} \in W$, it holds that

$$\mathop{\mathbf{E}}_{\mathrm{ALG}}\left[\max_{i\in[k]} |V_i \cap \tilde{E}|\right] \geq \mathbf{E}\left[\max_{i\in[k]} B'_i\right] \geq \frac{1}{12}\mathbf{E}\left[\max_{i\in[k]} B_i\right] \geq \frac{\mathrm{OPT}}{24},$$

where the first inequality holds because the values of the sampled multisets dominate the random variables B'_i. For the second inequality, intuitively, one can view the process of sampling the random variables B'_i as first sampling the Binomials $B_1, \ldots, B_k$ and then discarding every Bernoulli random variable that succeeded, independently, with probability $1/12$. In this way, for every outcome of the Binomials $B_1, \ldots B_k$, the Binomials $B'_1, \ldots, B'_k$ will retrieve, on expectation, a $1/12$ factor of their corresponding random variables B_i. Finally, for the third inequality we used Lemma 4. ◀

We continue by proving that for any "good" outcome $\tilde{E} \in W$, the expected value of the maximum of the sets $S_1, \ldots, S_k$, that consist of the unique elements of the sampled multisets, V_i, is still $\Theta(1) \cdot \mathrm{OPT}$.

▶ **Lemma 7.** *For any $\tilde{E} \in W$, it holds that*

$$\mathop{\mathbf{E}}_{\mathrm{ALG}}\left[\max_{i\in[k]} |S_i \cap \tilde{E}|\right] \geq \frac{e}{240(e-1)} \cdot \mathrm{OPT}.$$

Proof of Lemma 7. In order to analyze the expected value of the portfolio for the events $\tilde{E} \in W$, we will condition on the value of the maximum of the multisets $V_1, \ldots, V_k$. By the law of total expectation we get that

$$\mathop{\mathbf{E}}_{\mathrm{ALG}}\left[\max_{i\in[k]} |S_i \cap \tilde{E}|\right] = \sum_x \mathop{\mathbf{E}}_{\mathrm{ALG}}\left[\max_{i\in[k]} |S_i \cap \tilde{E}| \,\middle|\, \max_{i\in[k]} |V_i \cap \tilde{E}| = x\right] \cdot \mathop{\mathbf{Pr}}_{\mathrm{ALG}}\left[\max_{i\in[k]} |V_i \cap \tilde{E}| = x\right]. \quad (\ddagger)$$

Lower bounding the following expression,

$$\mathop{\mathbf{E}}_{\mathrm{ALG}}\left[\max_{i\in[k]} |S_i \cap \tilde{E}| \,\middle|\, \max_{i\in[k]} |V_i \cap \tilde{E}| = x\right],$$

can be seen as a balls and bins question. Specifically, we know that one of the k solutions formed by the algorithm sampled x items from $|\tilde{E} \cap H|$. Since the algorithm is sampling elements uniformly at random, those x items are also distributed uniformly at random inside $|\tilde{E} \cap H|$. We are interested in calculating the expected number of distinct items that were sampled. This question is equivalent to counting the non-empty bins after throwing x balls uniformly into $|\tilde{E} \cap H|$ bins. Using a standard result about the expected number of non-empty bins, we get that

$$\mathop{\mathbf{E}}_{\mathrm{ALG}}\left[\max_{i\in[k]} |S_i \cap \tilde{E}| \,\middle|\, \max_{i\in[k]} |V_i \cap \tilde{E}| = x\right] \geq \min\left\{\frac{x}{2}, \frac{3|\tilde{E} \cap H|}{10}\right\}.$$

Eq. ‡ can now be re-written as

$$\begin{aligned}
\mathop{\mathbf{E}}_{\mathrm{ALG}}\left[\max_{i\in[k]} |S_i \cap \tilde{E}|\right] &\geq \sum_x \min\left\{\frac{x}{2}, \frac{3|\tilde{E} \cap H|}{10}\right\} \cdot \mathop{\mathbf{Pr}}_{\mathrm{ALG}}\left[\max_{i\in[k]} |V_i \cap \tilde{E}| = x\right] && (1)\\
&\geq \sum_x \min\left\{\frac{x}{2}, \frac{\mathrm{OPT}}{40}\right\} \cdot \mathop{\mathbf{Pr}}_{\mathrm{ALG}}\left[\max_{i\in[k]} |V_i \cap \tilde{E}| = x\right] && (2)\\
&\geq \min\left\{\frac{\mathrm{OPT}}{300}, \frac{\mathrm{OPT}}{40}\right\} \cdot \mathop{\mathbf{Pr}}_{\mathrm{ALG}}\left[\max_{i\in[k]} |V_i \cap \tilde{E}| \geq \frac{\mathrm{OPT}}{150}\right] && (3)\\
&\geq \frac{\mathrm{OPT}}{300} \cdot \mathop{\mathbf{Pr}}_{\mathrm{ALG}}\left[\max_{i\in[k]} |V_i \cap \tilde{E}| \geq \frac{\mathbf{E}_{\mathrm{ALG}}\left[\max_{i\in[k]} |V_i \cap \tilde{E}|\right]}{5}\right] && (4)\\
&\geq \frac{e}{300(e-1)} \cdot \mathrm{OPT}, && (5)
\end{aligned}$$

where for (2) we used the fact that $|\tilde{E} \cap H| \geq \mathrm{OPT}/12$ for $\tilde{E} \in W$, to get (3) we restricted the sum to the terms $x \geq \mathrm{OPT}/150$, to get (4) we used that $\mathbf{E}_{\mathrm{ALG}}\left[\max_{i\in[k]} |V_i \cap \tilde{E}|\right] \geq \mathrm{OPT}/24$ from Lemma 6 and to get (5) we used a concentration inequality for the maximum of independent Binomial random variables and the assumption that $\mathrm{OPT} \geq 800$ to get that $\mathbf{E}_{\mathrm{ALG}}\left[\max_{i\in[k]} |V_i \cap \tilde{E}|\right]$ is at least some constant. ◀

Finally, we are ready to prove the main theorem of the section.

▶ **Theorem 8.** *Algorithm 1 is a $\Theta(1)$-approximation algorithm for the k-solution portfolio problem when the given matroid is uniform and $\mathcal{D}$ is a product distribution.*

Proof of Theorem 8. The value of the constructed portfolio $\mathcal{P}$ can be written as:

$$\begin{aligned}
\text{value}(\mathcal{P}) &= \mathop{\mathbf{E}}_{\text{ALG},A\sim\mathcal{D}}\left[\max_{i\in[k]}|S_i\cap A|\right] && (1)\\
&\geq \sum_{\tilde{E}\in W}\mathop{\mathbf{E}}_{\text{ALG}}\left[\max_{i\in[k]}|S_i\cap\tilde{E}|\right]\cdot\mathop{\mathbf{Pr}}_{A\sim\mathcal{D}}\left[A=\tilde{E}\right] && (2)\\
&\geq \frac{e}{300(e-1)}\cdot\text{OPT}\cdot\sum_{\tilde{E}\in W}\mathop{\mathbf{Pr}}_{A\sim\mathcal{D}}\left[A=\tilde{E}\right] && (3)\\
&\geq \frac{e}{300(e-1)}\cdot\text{OPT}\cdot\mathop{\mathbf{Pr}}_{A\sim\mathcal{D}}\left[A\in W\right] && (4)\\
&\geq \frac{e}{600(e-1)}\cdot\text{OPT}, && (5)
\end{aligned}$$

where to get (3) we applied Lemma 7 and (5) follows from Lemma 5. ◀

References

1 Maria-Florina Balcan. Data-driven algorithm design, 2020. arXiv:2011.07177.

2 Eric Balkanski, Baharan Mirzasoleiman, Andreas Krause, and Yaron Singer. Learning sparse combinatorial representations via two-stage submodular maximization. In *International Conference on Machine Learning*, pages 2207–2216. PMLR, 2016. URL: http://proceedings.mlr.press/v48/balkanski16.html.

3 Justin Chen, Sandeep Silwal, Ali Vakilian, and Fred Zhang. Faster fundamental graph algorithms via learned predictions. In *International Conference on Machine Learning*, pages 3583–3602. PMLR, 2022. URL: https://proceedings.mlr.press/v162/chen22v.html.

4 Sami Davies, Benjamin Moseley, Sergei Vassilvitskii, and Yuyan Wang. Predictive flows for faster ford-fulkerson. In *International Conference on Machine Learning*, pages 7231–7248. PMLR, 2023. URL: https://proceedings.mlr.press/v202/davies23b.html.

5 Sami Davies, Sergei Vassilvitskii, and Yuyan Wang. Warm-starting push-relabel. *arXiv preprint*, 2024. doi:10.48550/arXiv.2405.18568.

6 Jeff Decary, David Bergman, Carlos Cardonha, Jason Imbrogno, and Andrea Lodi. The madness of multiple entries in march madness, 2024. doi:10.48550/arXiv.2407.13438.

7 Michael Dinitz, Sungjin Im, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. Faster matchings via learned duals. *Advances in neural information processing systems*, 34:10393–10406, 2021. URL: https://proceedings.neurips.cc/paper/2021/hash/5616060fb8ae85d93f334e7267307664-Abstract.html.

8 Swati Gupta, Jai Moondra, and Mohit Singh. Balancing notions of equity: Approximation algorithms for fair portfolio of solutions in combinatorial optimization. *arXiv preprint*, 2023. doi:10.48550/arXiv.2311.03230.

9 Jon Kleinberg, Christos Papadimitriou, and Prabhakar Raghavan. Segmentation problems. *Journal of the ACM (JACM)*, 51(2):263–280, 2004. doi:10.1145/972639.972644.

10 Jon Kleinberg and Maithra Raghu. Team performance with test scores. *ACM Transactions on Economics and Computation (TEAC)*, 6(3-4):1–26, 2018. doi:10.1145/3274644.

11 Silvio Lattanzi, Ola Svensson, and Sergei Vassilvitskii. Speeding up bellman ford via minimum violation permutations. In *International Conference on Machine Learning*, pages 18584–18598. PMLR, 2023. URL: https://proceedings.mlr.press/v202/lattanzi23a.html.

12 Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. *Communications of the ACM*, 65(7):33–35, 2022. doi:10.1145/3528087.

13 George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions – I. *Mathematical programming*, 14:265–294, 1978. doi:10.1007/BF01588971.

14 Sahil Singla. Combinatorial optimization under uncertainty: Probing and stopping-time algorithms. *Unpublished doctoral dissertation, Carnegie Mellon University*, 2018.

15 Serban Stan, Morteza Zadimoghaddam, Andreas Krause, and Amin Karbasi. Probabilistic submodular maximization in sub-linear time. In *International Conference on Machine Learning*, pages 3241–3250. PMLR, 2017. URL: http://proceedings.mlr.press/v70/stan17a.html.

Concentration of Submodular Functions and Read-k Families Under Negative Dependence

Sharmila Duppala
University of Maryland, College Park, MD, USA

George Z. Li
Carnegie Mellon University, Pittsburgh, PA, USA

Juan Luque
University of Maryland, College Park, MD, USA

Aravind Srinivasan
University of Maryland, College Park, MD, USA

Renata Valieva
University of Maryland, College Park, MD, USA

Abstract

We study the question of whether submodular functions of random variables satisfying various notions of negative dependence satisfy Chernoff-like concentration inequalities. We prove such a concentration inequality for the lower tail when the random variables satisfy negative association or negative regression, partially resolving an open problem raised in ([23]). Previous work showed such concentration results for random variables that come from specific dependent-rounding algorithms ([6, 14]). We discuss some applications of our results to combinatorial optimization and beyond. We also show applications to the concentration of read-k families [13] under certain forms of negative dependence; we further show a simplified proof of the entropy-method approach of [13].

2012 ACM Subject Classification Mathematics of computing → Probabilistic algorithms

Keywords and phrases Chernoff bounds, Submodular Functions, Negative Correlation

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.47

Funding All authors are supported in part by NSF award number CCF-1918749.

Acknowledgements We thank the ITCS 2025 reviewers for their thoughtful comments.

1 Introduction

Concentration inequalities are ubiquitous in discrete mathematics and theoretical computer science [2, 9]. The most canonical examples are the Chernoff-Hoeffding bounds, which show strong concentration for linear combinations of independent random variables [7, 15]. In some applications, the condition of independence is too restrictive, so weaker notions have been considered [3, 24, 25]. Of interest to us is the setting where the random variables are negatively correlated, which arises naturally, for example, in designing approximation algorithms by solving a linear or semidefinite program and applying some dependent randomized rounding algorithm [11]. For this setting, [20] showed that the Chernoff-Hoeffding bounds can be shown under the weak notion of *negative cylinder dependence*: this and other standard notions of negative dependence are defined in Section 2.1.

For some applications in combinatorial optimization, algorithmic game theory, and machine learning, one needs to consider the more general class of *submodular functions* f of the random variables, rather than simple linear combinations. When the binary random variables $X_1, \ldots, X_n$ are independent, it was shown that $f(X_1, \ldots, X_n)$ still satisfies Chernoff


© Sharmila Duppala, George Z. Li, Juan Luque, Aravind Srinivasan, and Renata Valieva;
licensed under Creative Commons License CC-BY 4.0



bounds exactly [6]. When there is dependence between the random variables, the results are much weaker. The only known results are for random variables that are output by specific dependent-rounding algorithms, known as swap rounding and pipage rounding [6, 14]. These results showed that a Chernoff-like lower-tail bound also holds for submodular functions for their specific dependent rounding procedure. As noted in the work of [12], it is not clear how to generalize either of these proofs to any general notion of negative dependence.

We introduce a new notion of negative dependence, called 1-negative association, which is weaker than negative association and negative regression but stronger than negative cylinder dependence.

▶ **Definition 1.** *A collection of random variables $X_1, \ldots, X_n$ is said to satisfy 1-negative association if for any two monotone functions f and g, where g depends on a single random variable X_i and f depends on the remaining random variables $\{X_j\}_{j\in[n]\setminus\{i\}}$, we have $\mathbb{E}[fg] \leq \mathbb{E}[f]\mathbb{E}[g]$.*

Importantly, while in general it is weaker than the notion of weak negative regression introduced by [23], 1-negative association is equivalent to it when the variables $X_1, \ldots, X_n$ are binary. Further details are provided in Section 3.1.

Our main result is that the Chernoff-like bound shown in [6, 14] also hold under 1-negative association (see Section 3.2). In particular, this implies the following:

▶ **Theorem 2.** *Let $X_1, \ldots, X_n$ be binary random variables with mean $x_1, \ldots, x_n$ satisfying negative association (or negative regression). Let f be a non-negative monotone submodular function with marginal values in $[0, 1]$ and let F be the multilinear extension of f. If we let $\mu_0 = F(x_1, \ldots, x_n)$, then we have the following:*

$$\Pr[f(X_1, \ldots, X_n) \leq (1-\delta)\cdot\mu_0] \leq \exp(-\mu_0\delta^2/2).$$

A few remarks are in order. First, we highlight that the concentration in the above theorem is with respect to the value of the multilinear extension $F(x_1, \ldots, x_n)$, rather than the true expected value $\mathbb{E}[f(X_1, \ldots, X_n)]$. In general, the true expected value can be greater than the value of multilinear extension [23]. Nevertheless, this suffices for applications relating to submodular maximization, and is the same type of concentration result shown in previous work. Second, recall that negative cylinder dependence does not suffice to show this concentration bound [6, p. 583]. As a result, our results are, in some informal sense, almost tight in terms of the condition on negative dependence.

In addition to providing submodular concentration results for a wide class of rounding algorithms and distributions, our results also give a new path toward understanding why pipage rounding and swap rounding satisfy the lower-tail Chernoff bound. By proving that the rounding algorithms output random variables which are 1-negatively associated, we immediately obtain a new proof of the lower tail bounds. This can be viewed as evidence that the two rounding algorithms satisfy 1-negative association or even negative association/regression. We leave this as an interesting open question.

Techniques

We use the standard method of bounding the exponential moments for lower-tail Chernoff bounds. Our idea is to show that the exponential moments for our negatively-correlated random variables is upper bounded by the exponential moments for independent copies of the random variables. Formally, let $X_1, \ldots, X_n$ be random variables satisfying 1-negative association and let $X_1^*, \ldots, X_n^*$ be independent copies of the random variables. We show for any $\lambda < 0$, we have

$$\mathbb{E}[\exp(\lambda \cdot f(X_1, \ldots, X_n))] \leq \mathbb{E}[\exp(\lambda \cdot f(X_1^*, \ldots, X_n^*))].$$

Since the exponential-moments method has been used to prove Chernoff bounds for submodular functions in the independent case [6], we can then repeat their proof and conclude with our desired result. We believe this proof idea may be of independent interest. For example, the same ideas can show that for a supermodular function g and any $\lambda > 0$, we have

$$\mathbb{E}[\exp(\lambda \cdot g(X_1, \ldots, X_n))] \leq \mathbb{E}[\exp(\lambda \cdot g(X_1^*, \ldots, X_n^*))].$$

In other words, we have morally proven the following statement: any upper-tail concentration bound which can be proven for a supermodular function g under independence based on the exponential-moments method also holds when the underlying random variables are negatively associated. As an example, we can apply this to a read-k family of supermodular functions $g_1, \ldots, g_r$ for negatively associated random variables [13]. A read-k family is defined as a set of functions where each variable appears in at most k functions. This concept is particularly useful in scenarios where functions need to model or manage overlapping sets of variables with constraints on their interaction. We highlight that the proof of concentration for read-k families given in [13] doesn't use the exponential-moments method, but instead it is based on the entropy method. We address this by giving a simpler proof of their results, this time using the exponential moments method. This gives the first concentration results for a class of supermodular functions under negatively correlated random variables, and is detailed in Section 3.3.

Applications

Our motivation for studying the problem comes from the randomized-rounding paradigm in approximation algorithms for converting a fractional solution to a linear program into an integral one. In many such randomized-rounding schemes, the output random variables have been shown to satisfy strong negative dependence properties, such as negative association [26, 11]. For all such rounding algorithms, our results immediately imply the submodular Chernoff lower-tail bound. It remains an interesting open question to efficiently sample negatively dependent distributions for a wider class of set systems. A particularly interesting algorithm is given in the work of [22]; they show that a fractional point in a matroid polytope can be rounded to an integral one such that the resulting distribution preserves marginals and satisfies negative association. However, a gap identified in their proof [23] complicates the application of their approach. The implications of this issue for the applicability of our results remain an area for further investigation.

As a concrete application, we consider the maximum coverage problem under group fairness constraints. Here, we have a universe of elements $\{1, \ldots, n\}$, a collection $S_1, \ldots, S_m$ of subsets of the universe, and a budget k. We are further given subsets $C_1, \ldots, C_\ell \subseteq [n]$ (which should be thought of as demographic groups) along with thresholds $w_1, \ldots, w_\ell$. Our goal is to choose k sets from the collection to maximize the number of elements covered subject to the fairness constraint each demographic group is sufficiently covered (i.e., at least w_j elements from C_j are covered). Since this is a special case of multiobjective submodular maximization, there exists a $(1 - 1/e - \epsilon)$-approximation to the problem such that each fairness constraint is approximately satisfied [6, 28]. Unfortunately, these results rely on the randomized swap-rounding algorithm due to its submodular concentration properties, which requires a super-linear time complexity. While swap rounding can be implemented with poly-logarithmic depth [5], a simpler dependent-rounding algorithm of [26] requires linear work and only $O(\log n)$ depth, which improves the efficiency. Observe that the pre-processing step in [28] only requires $O(n\ell)$ time. Since we can solve the linear program for fair maximum

coverage in near-linear time [1], we obtain a near-linear time algorithm for the problem after using the efficient rounding algorithm of [26]. These same ideas can be used to improve the time complexity of the algorithm by [27] for influence maximization with group-fairness constraints. Since the proofs are similar to previous work, we defer the details to a future version of the paper.

More generally, negatively-associated random variables show up naturally in many settings (see e.g., the primer by [29]). [8] studied the canonical example of *balls and bins*, and showed that it satisfied both negative association and negative regression. Another example satisfying the negative-association conditions are any product measure over the set of bases of a balanced matroid, as shown by [10]. A final setting where such random variables occur are random spanning trees, which have been vital in the recent improvements to approximation algorithms for the traveling salesperson problem (see, e.g., [16]). Random spanning trees are known to be strongly Rayleigh, which immediately implies that they are negatively associated. Our results may be interesting here as well.

We also observe that the online rounding scheme of [18] has the strongly Rayleigh property: we immediately get strong concentration (on the lower-tail side) for monotone submodular functions, when the inputs for the function arrive online along with their (Bernoulli) distributions as in the setup of [18].

Related Work

The concentration of negatively-dependent random variables was first formally studied by [19], which showed a central limit theorem for a certain notion of negative dependence. Later on, [20] showed that cylinder negatively dependent random variables yield the Chernoff-Hoeffding concentration inequalities, just like independent random variables. In the context of our paper, these results are somewhat specialized since they focus on linear combinations of random variables.

For non-linear functions of the random variables, the majority of work has focused on the concentration of Lipschitz functions under various notions of negative dependence. [21] showed that for strong Rayleigh measures, one has Gaussian concentration for any Lipschitz function. Later on, [12] corrected an earlier proof of [8], showing that McDiarmid-like concentration results hold for Lipschitz functions of random variables satisfying negative regression. These results are complementary to ours since we are trying to give dimension-free concentration results.

2 Preliminaries

2.1 Notions of Negative Dependence

We begin by defining the notion of negative dependence commonly found in the literature.

Negative Cylinder Dependence

A collection of Boolean random variables $X_1, \ldots, X_n$ is said to be negative cylinder dependent if for every $S \subseteq [n]$,

$$\mathbb{E}\left[\prod_{i \in S} X_i\right] \leq \prod_{i \in S} \mathbb{E}\left[X_i\right]$$

and

$$\mathbb{E}\left[\prod_{i \in S} (1 - X_i)\right] \leq \prod_{i \in S} \mathbb{E}\left[1 - X_i\right].$$

Negative cylinder dependence is the weaker notion considered here. It is known to imply Chernoff bounds for linear combinations of $X_1, \ldots, X_n$ but it is insufficient to show our submodular concentration results.

Negative Association

A collection of random variables $X_1, \ldots, X_n$ is said to be negatively associated if for any $I, J \subset [n], I \cap J = \emptyset$ and any pair of non-decreasing functions $f : \mathbb{R}^I \to \mathbb{R}, g : \mathbb{R}^J \to \mathbb{R}$,

$$\mathbb{E}\left[f\left(X_I\right) g\left(X_J\right)\right] \leq \mathbb{E}\left[f\left(X_I\right)\right] \mathbb{E}\left[g\left(X_J\right)\right].$$

Here and in the following, X_S refers to those random variables that are indexed by the elements in S, $X_S = \{X_i \ : \ i \in S\}$. Negative association is a significant strengthening of negative cylinder dependence, and has many additional useful closure properties. This will be one of the focuses of the paper.

Negative Regression

A collection of random variables $X_1, \ldots, X_n$ is said to satisfy negative regression, if for any $I, J \subset [n], I \cap J = \emptyset$, any non-decreasing function $f : \mathbb{R}^I \to \mathbb{R}$ and $a \leq b \in \mathbb{R}^J$,

$$\mathbb{E}\left[f\left(X_I\right) \mid X_J = a\right] \geq \mathbb{E}\left[f\left(X_I\right) \mid X_J = b\right].$$

Negative regression is a strengthening of negative cylinder dependence, but its relationship with negative association is not yet well understood. It is known that negative association doesn't imply negative regression [8], but the opposite implication is not known. This will be the other focus of the paper.

Strong Rayleigh

A collection of random variables $X_1, \ldots, X_n$ is said to satisfy the strong Rayleigh property if the generating function

$$F(z_1, \ldots, z_n) = \mathbb{E}[\textstyle\prod_{j=1}^{n} z_j^{X_j}]$$

is a real stable polynomial (i.e., it has no root $(z_1, \ldots, z_n) \in \mathbb{C}^n$ with all positive imaginary components). The strong Rayleigh property is the strongest notion of negative dependence, and has been shown to imply all other studied negative dependence definitions [4]. As a result, all of our results apply here as well.

2.2 Submodular Functions

We also give a quick review of the basics of submodular functions.

Submodular Functions

We say that a function $f : \{0,1\}^n \to \mathbb{R}$ is submodular if

$$f(X_1, \ldots, X_{i-1}, 1, X_{i+1}, \ldots X_n) - f(X_1, \ldots, X_{i-1}, 0, X_{i+1}, \ldots X_n)$$

is a non-increasing function of $X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n$ for each $i \in [n]$. When viewing the binary input of f as the indicator vector for a set, this is equivalent to the more common definition that f is submodular if for any $X, Y \subseteq [n]$ with $X \subseteq Y$ and any $x \notin Y$, we have

$$f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y).$$

Supermodular Functions

We say that a function $g : \{0,1\}^n \to \mathbb{R}$ is supermodular if

$$g(X_1, \ldots, X_{i-1}, 1, X_{i+1}, \ldots X_n) - g(X_1, \ldots, X_{i-1}, 0, X_{i+1}, \ldots X_n)$$

is a non-decreasing function of $X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n$ for each $i \in [n]$. When viewing the binary input of g as the indicator vector for a set, this is equivalent to the more common definition that g is supermodular if for any $X, Y \subseteq [n]$ with $X \subseteq Y$ and any $x \notin Y$, we have

$$g(X \cup \{x\}) - g(X) \leq g(Y \cup \{x\}) - g(Y).$$

Mutlilinear Extension

The multilinear extension of a function f is

$$F(x) = \mathbb{E}[f(x)] = \sum_{S \subseteq N} f(S) \prod_{i \in S} x_i \prod_{i \notin S} (1 - x_i),$$

for $x \in [0,1]^n$. If we view x as a probability vector, the multilinear extension F is simply the expected value of f when each coordinate is rounded independently in $\{0,1\}$.

3 Submodular Chernoff Bounds

3.1 1-Negative Association and Weak Negative Regression

We first define the weaker notion of negative dependence which we work with, called 1-negative association, and prove some simple properties about it. We also define a related notion of weak negative regression, which is the analogue of 1-negative association for the notion of negative regression, and we show the equivalence between the two for binary random variables and show that weak negative regression is strictly stronger in general. After an initial draft, we discovered that [23] had already introduced the notion of weak negative regression for binary random variables in a context complementary to ours. Using their work, we can immediately show nice properties about 1-negative association.

▶ **Definition 3.** *A collection of random variables $X_1, \ldots, X_n$ is said to satisfy 1-negative association if for any two monotone functions f and g, where g depends on a single random variable X_i and f depends on the remaining random variables $\{X_j\}_{j \in [n] \setminus \{i\}}$, we have $\mathbb{E}[fg] \leq \mathbb{E}[f]\mathbb{E}[g]$.*

▶ **Definition 4.** *A collection of random variables $X_1, \ldots, X_n$ is said to satisfy weak negative regression if for any index i and any monotone function f depending on the remaining random variables $\{X_j\}_{j \in [n] \setminus \{i\}}$, we have $\mathbb{E}[f | X_i = b] \leq \mathbb{E}[f | X_i = a]$ for all $a \leq b$.*

In the following lemmata, we show that weak negative regression implies 1-negative association in general. We then show that the reverse implication holds for binary random variables, but give an example showing that it does not hold in general.

▷ Claim 5. If a collection of random variables $X_1, \ldots, X_n$ satisfies weak negative regression, then it satisfies 1-negative association.

Proof. Assume $X_1, \ldots, X_n$ satisfy weak negative regression; we will prove that it also satisfies 1-negative association. Let f and g be monotone functions such that f depends on X_I for some subset $I \subseteq [n]$ and g depends on X_i for $i \notin I$. Without loss of generality, let us assume that f and g are non-decreasing.

First, define the function $F(a) = \mathbb{E}[f(X)|X_i = a]$. Evaluating the expectation $\mathbb{E}[f(X_I)g(X_i)]$, we can express it via the law of total expectation as

$$\mathbb{E}[f(X_I)g(X_i)] = \mathbb{E}[\mathbb{E}[f(X_I)|X_i]g(X_i)] = \mathbb{E}[F(X_i)g(X_i)].$$

Next, we observe that by the definition of weak negative regression, $F(a)$ is non-increasing. Therefore, random variables $F(X_i)$ and $g(X_i)$ are negatively correlated, yielding

$$\mathbb{E}[F(X_i)g(X_i)] \leq \mathbb{E}[F(X_i)]\mathbb{E}[g(X_i)].$$

Converting $\mathbb{E}[F(X_i)]$ back into the terms of f and g, we find

$$\mathbb{E}[F(X_i)]\mathbb{E}[g(X_i)] = \mathbb{E}[f(X)]\mathbb{E}[g(X_i)].$$

The overall inequality $\mathbb{E}[f(X_I)g(X_i)] \leq \mathbb{E}[f(X_I)]\mathbb{E}[g(X_i)]$ thus holds, which establishes that $X_1, \ldots, X_n$ are 1-negatively associated, completing the proof. ◁

▷ Claim 6. If a collection of binary random variables $X_1, \ldots, X_n$ satisfies 1-negative association, then it satisfies weak negative regression.

Proof. Recall that we wish to prove that for any non-decreasing function f depending on some subset $I \subseteq [n]$ and any $i \notin I$, we have that

$$\mathbb{E}[f(X_I)|X_i = 0] \geq \mathbb{E}[f(X_I)|X_i = 1].$$

By the definition of 1-negative association, we obtain that for any monotone functions g which depends on X_i, the following inequality holds:

$$\mathbb{E}[f(X_I)g(X_i)] \leq \mathbb{E}[f(X_I)] \cdot \mathbb{E}[g(X_i)]. \tag{1}$$

Without loss of generality, we may assume that $\mathbb{E}[f(X_I)|X_i = 1] = 0$ by shifting f by a constant. By choosing g to be the identity function, we can apply the law of total probability to obtain

$$\mathbb{E}[f(X_I) \cdot g(X_i)] = \Pr[X_i = 0] \cdot \mathbb{E}[f(X_I)|X_i = 0] \cdot g(0) + \Pr[X_i = 1] \cdot \mathbb{E}[f(X_I)|X_i = 1] \cdot g(1) = 0.$$

Plugging this into Equation 1, we obtain

$$0 \leq \mathbb{E}[f(X_I)] \cdot \mathbb{E}[g(X_i)] = \mathbb{E}[f(X_I)|X_i = 0] \Pr[X_i = 0] \cdot \mathbb{E}[X_i],$$

again by the law of total probability. Since $\mathbb{E}[X_i] > 0$ and $\Pr[X_i = 0] > 0$, this implies that

$$\mathbb{E}[f(X_I)|X_i = 0] \geq 0,$$

which concludes the proof since $\mathbb{E}[f(X_I)|X_i = 1] = 0$. ◁

▷ Claim 7. There exists (non-binary) distributions over 2 random variables which satisfy 1-negative association but not weak negative regression. In other words, 1-negative association is strictly more general than weak negative regression for non-binary random variables.

Proof. Let's first discuss the intuition for the construction of the counterexample. One can show via algebra that $\mathbb{E}[f(X_I)g(X_i)] - \mathbb{E}[f(X_I)]\mathbb{E}[g(X_i)]$ can be expanded as the following expression:

$$\textstyle\sum_{x<y} \Pr[X_i = x] \Pr[X_i = y] \cdot \big(\mathbb{E}[f(X_I)|X_i = x] - \mathbb{E}[f(X_I)|X_i = y]\big)\big(g(x) - g(y)\big),$$

where the summation is over the values that X_i takes with non-zero probability.

Now, suppose that for some values $x < y$ we have that

$$\mathbb{E}[f(X_1, \ldots, X_n)|X_i = x] - \mathbb{E}[f(X_1, \ldots, X_n)|X_i = y] < 0,$$

violating the weak negative regression property. This would imply that some of the summands are positive (since $g(x) < g(y)$ by monotonicity). In the case of binary random variables, the summation would only consist of a single summand so 1-negative association would be violated. For general random variables, the summation consists of multiple terms so the summation may still be negative even when a single summand is positive. Consequently, the random variables may still satisfy 1-negative association.

We now give the example. Consider the random variables (X_1, X_2) which are uniformly distributed on their support set $\{(0,3),(1,1),(2,2),(3,0)\}$. By considering an identity function $\mathbb{1}_x : \{0,1,2,3\} \to \{0,1,2,3\}$, we can show that that the distribution of (X_1, X_2) does not satisfy weak negative regression:

$$\mathbb{E}[\mathbb{1}_x(X_2)|X_1 = 1] = \mathbb{1}_x(1) = 1 < 2 = \mathbb{1}_x(2) = \mathbb{E}[\mathbb{1}_x(X_2)|X_1 = 2].$$

However, for any pair of non-decreasing functions $f, g : \{0,1,2,3\} \to \mathbb{R}$, we have

$$\mathbb{E}[f(X_1)]\mathbb{E}[g(X_2)] - \mathbb{E}[f(X_1)g(X_2)] = \frac{f(1)+f(2)+f(3)}{4} \cdot \frac{g(1)+g(2)+g(3)}{4} - \frac{f(1)g(1)+f(2)g(2)}{4},$$

where we have again assumed without loss of generality that $f(0) = g(0) = 0$.

We claim that the quantity on the right hand side is always non-negative. In order to see this, observe that $f(2)g(2) \leq f(i)g(j)$ for any $i, j \geq 2$ by monotonicity. As a result, we have

$$\frac{f(2)g(2)}{4} \leq \frac{(f(2)+f(3))(g(2)+g(3))}{16}.$$

Further, we observe that $f(1)g(1) \leq f(i)g(j)$ for any $i, j \geq 1$ by monotonicity. As a result, we have

$$\frac{f(1)g(1)}{4} \leq \frac{f(1)(g(2)+g(3)) + g(1)(f(2)+f(3))}{16}.$$

Combining these two inequalities immediately and observing that $f(1)g(1) \geq 0$ by monotonicity implies our desired result. Hence, the distribution is 1-negatively associated. ◁

Since 1-negative association and weak negative regression are equivalent for binary random variables and weak negative regression has been shown to be strictly stronger than cylinder negative dependence [23, Proposition 2.4], we also have that 1-negative association is strictly stronger than cylinder negative dependence. Additionally, since weak negative regression is strictly stronger than 1-negative association for general random variables and weak negative regression has been shown to be strictly weaker than negative association and negative regression [23, Proposition 2.4], we have that 1-negative association is strictly weaker than negative association and negative regression. We summarize these in the following corollaries.

▶ **Corollary 8.** *1-negative association is a strictly weaker condition than negative association.*

▶ **Corollary 9.** *1-negative association is a strictly weaker condition than negative regression.*

▶ **Corollary 10.** *1-negative association is a strictly stronger condition than negative cylinder dependence.*

3.2 Proof of Submodular Concentration

We will now prove our main result. As mentioned in the introduction, our proof is based on the standard technique of bounding the exponential moments. The following lemma contains our main technical contribution, stating that the exponential moments of $f(X_1, \ldots, X_n)$ under 1-negative association is dominated by that under independence. Our results will follow easily afterwards.

▶ **Lemma 11.** *Let $X_1, \ldots, X_n$ be 1-negatively associated random variables and let $X_1^*, \ldots, X_n^*$ be independent random variables with the same marginal distributions. Also let f be a non-negative monotone function.*

- *If f is submodular and $\lambda < 0$, we have* $\mathbb{E}[\exp(\lambda f(X_1, \ldots, X_n))] \leq \mathbb{E}[\exp(\lambda f(X_1^*, \ldots, X_n^*))]$.
- *If f is supermodular and $\lambda > 0$, we have* $\mathbb{E}[\exp(\lambda f(X_1, \ldots, X_n))] \leq \mathbb{E}[\exp(\lambda f(X_1^*, \ldots, X_n^*))]$.

Proof. Fix $\lambda < 0$ if f is submodular and $\lambda > 0$ if f is supermodular. Observe that in order to prove the lemma, it suffices to prove

$$\mathbb{E}[\exp(\lambda \cdot f(X_1, \ldots, X_i, \ldots, X_m))] \leq \mathbb{E}[\exp(\lambda \cdot f(X_1, \ldots, X_i^*, \ldots, X_m))], \tag{2}$$

since we can iteratively apply the above inequality to each X_i (note that we can do this because independent variables are also negatively associated). For simplicity of notation and without loss of generality, we will prove the inequality for $i = 1$.

By considering the cases of $X_1 = 0$ and $X_1 = 1$ separately, we have

$$\exp(\lambda \cdot f(X_1, \ldots, X_n)) = X_1 \cdot \exp(\lambda \cdot f(1, X_2, \ldots, X_n))) + (1 - X_1) \cdot \exp(\lambda \cdot f(0, X_2, \ldots, X_n))),$$

where the equality holds pointwise on the underlying probability space. Via simple algebraic manipulations, we can rewrite the right hand side as

$$X_1 \cdot [\exp(\lambda \cdot f(1, X_2, \ldots, X_n))) - \exp(\lambda \cdot f(0, X_2, \ldots, X_n))] + \exp(\lambda \cdot f(0, X_2, \ldots, X_n))$$

Taking expectations, we now have that $\mathbb{E}[\exp(\lambda f(X_1, \ldots, X_n))]$ can be written as

$$\mathbb{E}\big[X_1 \cdot [\exp(\lambda \cdot f(1, X_2, \ldots, X_n))) - \exp(\lambda \cdot f(0, X_2, \ldots, X_n))]\big] + \mathbb{E}\big[\exp(\lambda \cdot f(0, X_2, \ldots, X_n))\big] \tag{3}$$

Observe that X_1 is clearly an increasing function of X_1. We claim that if either (i) f is submodular and $\lambda < 0$ or (ii) f is supermodular and $\lambda > 0$, we have that $\exp(\lambda \cdot f(1, X_2, \ldots, X_n))) - \exp(\lambda \cdot f(0, X_2, \ldots, X_n))$ is an increasing function in $X_2, \ldots, X_n$. Indeed, we first rewrite the function as

$$\exp(\lambda \cdot f(0, X_2, \ldots, X_n)) \cdot [\exp(\lambda \cdot (f(1, X_2, \ldots, X_n) - f(0, X_2, \ldots, X_n))) - 1] =: A_1 \cdot A_2$$

for simplicity of notation.

Let us first consider the case when $\lambda < 0$ and f is submodular. We have that A_1 is (i) positive because the exponential function is always positive and (ii) non-increasing in $X_2, \ldots, X_n$ because f is non-decreasing and $\lambda < 0$. We also have that A_2 is (i) negative because the argument in $\exp(\cdot)$ is negative, so the exponential is in $(0, 1)$ (ii) non-decreasing since $\lambda < 0$ and the difference of f evaluated at $X_1 = 1$ and $X_1 = 0$ is non-increasing by definition of submodularity. Hence, our expression of interest is the product of a function A_1 which decreases towards 0 and a function A_2 which increases towards 0. The product will be negative and monotonically increasing towards 0.

Now, let us consider the case when $\lambda > 0$ and f is supermodular. We have that A_1 is (i) positive because the exponential function is always positive and (ii) non-decreasing since $\lambda > 0$, f is monotone, and $\exp(\cdot)$ is also monotone. We also have that A_2 is (i) positive because the argument of $\exp(\cdot)$ is positive since f is monotone so the exponential is greater than 1 and (ii) non-decreasing since $\lambda > 0$ and the difference of f evaluated at $X_1 = 1$ and $X_1 = 0$ is non-decreasing by definition of supermodularity. As a result, the product will be positive and non-decreasing, as desired.

Since we have shown that the A_1A_2 is also monotone, we now have that the first term in Equation 3 can be written as the product of monotone functions of disjoint subsets, one of which is the singleton set. By 1-negative association, we have that the first term is upper bounded by

$$\mathbb{E}[X_1] \cdot \mathbb{E}[\exp(\lambda \cdot f(1, X_2, \ldots, X_n)) - \exp(\lambda \cdot f(0, X_2, \ldots, X_n))].$$

Consequently, the entire expression in (3) is upper bounded by

$$\mathbb{E}[X_1] \cdot \mathbb{E}[\exp(\lambda \cdot f(1, X_2, \ldots, X_n)) - \exp(\lambda \cdot f(0, X_2, \ldots, X_n))] + \mathbb{E}[\exp(\lambda \cdot f(0, X_2, \ldots, X_n))].$$

Since X_1 and X_1^* have the same marginal distributions, the above is exactly equal to

$$\mathbb{E}[X_1^*] \cdot \mathbb{E}[\exp(\lambda \cdot f(1, X_2, \ldots, X_n)) - \exp(\lambda \cdot f(0, X_2, \ldots, X_n))] + \mathbb{E}[\exp(\lambda \cdot f(0, X_2, \ldots, X_n))].$$

And since X_1^* is independent with $X_2, \ldots, X_m$ by assumption, the above is equal to

$$\mathbb{E}[X_1^* \cdot \exp(\lambda \cdot f(1, X_2, \ldots, X_n)) - \exp(\lambda \cdot f(0, X_2, \ldots, X_n))] + \mathbb{E}[\exp(\lambda \cdot f(0, X_2, \ldots, X_n))].$$

In particular, observe that this is in the exact same form as Equation 3, except with X_1 replaced with X_1^*. Note that when we transformed the left-hand side of Equation 2 to Equation 3, we never used any properties of the random variables $X_1, \ldots, X_n$ other than the fact that they take values in $\{0, 1\}$. As a result, we can reverse the direction of all of the equalities to show that the above expression is equal to

$$\mathbb{E}[\exp(\lambda \cdot f(X_1^*, X_2, \ldots, X_n))],$$

which completes the proof of the lemma. ◀

Now, we will complete the proof of our main result. Combining the theorem below with Claims 8 and 9 immediately gives a proof of Theorem 2. Here, our proof will rely heavily on the proof of the Chernoff bound for submodular functions under independence given in [6].

▶ **Theorem 12.** *Let $X_1, \ldots, X_n$ be binary random variables with mean $x_1, \ldots, x_n$ satisfying 1-negative association. Let f be a non-negative monotone submodular function with marginal values in $[0, 1]$ and let F be the multilinear extension of f. If we let, $\mu_0 = F(x_1, \ldots, x_n)$, then we have the following:*

$$\Pr[f(X_1, \ldots, X_n) \leq (1 - \delta) \cdot \mu_0] \leq \exp(-\mu_0 \delta^2 / 2).$$

Proof. Let $X_1^*, \ldots, X_n^*$ be independent random variables with the same respective marginals as $X_1, \ldots, X_n$ and let $\lambda < 0$ be a parameter to be set later. Let us decompose $f(X_1^*, \ldots, X_n^*) = \sum_{i=1}^n Y_i^*$, where

$$Y_i^* = f(X_1^*, \ldots, X_i^*, 0, \ldots, 0) - f(X_1^*, \ldots, X_{i-1}^*, 0, \ldots, 0).$$

Let us denote $\mathbb{E}[Y_i^*] = \omega_i$ and $\mu_0 = \sum_{i=1}^n \omega_i = \mathbb{E}[f(X_1^*, \ldots, X_n^*)]$. By the convexity of the exponential and the fact that $Y_i^* \in [0,1]$, we have that

$$\mathbb{E}[\exp(\lambda \cdot Y_i^*)] \leq \omega_i \cdot \exp(\lambda) + (1-\omega_i) = 1 + [\exp(\lambda) - 1] \cdot \omega_i \leq \exp[(\exp(\lambda) - 1) \cdot \omega_i].$$

Combining the above with Lemma C.1 from [6], we have that

$$\begin{aligned}\mathbb{E}[\exp(\lambda \cdot f(X_1^*, \ldots, X_n^*)] = \mathbb{E}[\exp(\lambda \cdot \textstyle\sum_{i=1}^n Y_i^*)] &\leq \textstyle\prod_{i=1}^n \mathbb{E}[\exp(\lambda \cdot Y_i^*)] \\ &\leq \exp[(\exp(\lambda) - 1) \cdot \mu_0]. \end{aligned} \tag{4}$$

Now, we can follow the proof of the standard Chernoff bound:

$$\begin{aligned}\Pr[f(X_1, \ldots, X_n) \leq (1-\delta) \cdot \mu_0] &= \Pr[\exp(\lambda \cdot f(X_1, \ldots, X_n)) \leq \geq \exp(\lambda(1-\delta) \cdot \mu_0)] \\ &\leq \frac{\mathbb{E}[\exp(\lambda \cdot f(X_1, \ldots, X_n))]}{\exp(\lambda(1-\delta) \cdot \mu_0)} \\ &\leq \frac{\mathbb{E}[\exp(\lambda \cdot f(X_1^*, \ldots, X_n^*))]}{\exp(\lambda(1-\delta) \cdot \mu_0)} \\ &\leq \frac{\exp[(\exp(\lambda) - 1) \cdot \mu_0]}{\exp(\lambda(1-\delta) \cdot \mu_0)}\end{aligned}$$

The first equality follows since $\exp(\lambda \cdot x)$ is a monotone function, the first inequality follows by Markov's inequality, the second inequality follows by Lemma 8, and the final inequality follows Equation 4.

Finally, we can choose λ such that $\exp(\lambda) = 1 - \delta$, which gives

$$\Pr[f(X_1, \ldots, X_n) \geq (1-\delta)\mu_0] \leq \frac{\exp(-\delta\mu_0)}{(1-\delta)^{(1-\delta)\mu_0}} \leq \exp(-\mu_0 \cdot \delta^2/2),$$

where we used $(1-\delta)^{1-\delta} \leq \exp(-\delta + \delta^2/2)$ for $\delta \in (0,1]$ in the final inequality. ◀

3.3 Concentration of read-k families

In this subsection, we illustrate an application of our proof technique to give concentration for a read-k family of supermodular functions. Read-k families arise naturally in problems such as subgraph counting in random graphs, and can be seen as a complementary weak dependence notion to that of low-degree polynomials [17]. Our work gives the first concentration results for these problems under negative dependence.

Let's consider this notion of weak dependence defined in [13]. Let $Y_1, \ldots, Y_n$ be random variables and assume that they can be factored as functions of random variables $X_1, \ldots, X_m$. We say that $Y_1, \ldots, Y_n$ are a read-k family of $X_1, \ldots, X_m$ if for each variable X_i, there are at most k variables among $Y_1, \ldots, Y_n$ that are influenced by X_i. Formally, we have the following.

▶ **Definition 13.** *Let $X_1, \ldots, X_m$ be random variables. For each $j \in [n]$, let $P_j \subseteq [m]$ and let $f_j : \{0,1\}^{P_j} \to [0,1]$ be functions of X_{P_j}. We say that $Y_j = f_j(X_{P_j})$ are a read-k family if $|\{j : i \in P_j\}| \leq k$ for each $i \in [m]$ (i.e., each variable X_i influences at most k functions).*

When $X_1, \ldots, X_m$ are independent, [13] showed that we have

$$\Pr[\textstyle\sum_{j=1}^n f_j(X_{P_j}) \geq (p+\epsilon)n] \leq \exp(-D(p+\epsilon||p) \cdot n/k) \tag{5}$$

$$\Pr[\textstyle\sum_{j=1}^n f_j(X_{P_j}) \leq (p-\epsilon)n] \leq \exp(-D(p-\epsilon||p) \cdot n/k), \tag{6}$$

where $p = (1/n)\sum_{j=1}^{n} \mathbb{E}[Y_j]$ and $D(\cdot||\cdot)$ is the Kullback-Leibler divergence. Notably, while Gavinsky et al. do not require random variables $X_1, \ldots, X_m$ to be binary, as we do in our approach, their model requires $Y_1, \ldots, Y_n$ to be binary. We will show that Inequality 5 on the upper tail still holds for supermodular functions $f_1, \ldots, f_n$. Similarily, Inequality 6, which addresses the lower tail bound, continues to apply to submodular functions $f_1, \ldots, f_n$.

▶ **Theorem 14.** *Let $X_1, \ldots, X_m$ be 1-negatively associated random variables and let $X_1^*, \ldots, X_m^*$ be independent random variables with the same respective marginal distributions as $X_1, \ldots, X_m$. Suppose that $f_j(X_{P_j})$ for $j \in [n]$ are a read-k family, where $f_j : \{0,1\}^{P_j} \to [0,1]$ are supermodular functions. If we let $p_0 = (1/n)\sum_{j=1}^{n} \mathbb{E}[f_j(X_{P_j}^*)]$ denote the averaged expectation when the underlying random variables are independent, we have*

$$\Pr[\textstyle\sum_{j=1}^{n} f_j(X_{P_j}) \geq (p_0 + \epsilon)n] \leq \exp(-D(p_0 + \epsilon||p_0) \cdot n/k).$$

Proof. Let $f(X_1, \ldots, X_m) = \sum_{j=1}^{n} f_j(X_{P_j})$ be the quantity of interest, and note that f is the sum of supermodular functions so it is supermodular as well.

We will follow the standard proof via exponential moments. Let $\lambda > 0$; we have

$$\Pr[f(X_1, \ldots, X_m) \geq (p_0 + \epsilon)n] = \Pr[\exp(\lambda \cdot f(X_1, \ldots, X_m)) \geq \exp(\lambda \cdot (p_0 + \epsilon)n)] \tag{7}$$

$$\leq \mathbb{E}[\exp(\lambda \cdot f(X_1, \ldots, X_m))]/\exp(\lambda \cdot (p_0 + \epsilon)n), \tag{8}$$

where the inequality follows by Markov's. Since f is supermodular, we have by Lemma 11 that

$$\mathbb{E}[\exp(\lambda \cdot f(X_1, \ldots, X_m))] \leq \mathbb{E}[\exp(\lambda \cdot f(X_1^*, \ldots, X_m^*))] = \mathbb{E}[\exp(\lambda \cdot \textstyle\sum_{j=1}^{n} f_j(X_{P_j}^*))]. \tag{9}$$

In the new proof of concentration of read-k families, given in the Appendix, we show that

$$\mathbb{E}[\exp(\lambda \cdot \textstyle\sum_{j=1}^{n} f_j(X_{P_j}^*))] \leq \left(\prod_{j=1}^{n} \mathbb{E}[\exp(\lambda \cdot f_j(X_{P_j}^*))^k]\right)^{1/k}. \tag{10}$$

Combining equations 7–10, we have

$$\Pr[f(X_1, \ldots, X_m) \geq (p_0 + \epsilon)n] \leq \left(\prod_{j=1}^{n} \mathbb{E}[\exp(k\lambda \cdot f_j(X_{P_j}^*))/\exp(k\lambda(p_0 + \epsilon)n]\right)^{1/k}.$$

Let $\lambda' = k\lambda$; since $\lambda > 0$ is a parameter we set, we can view $\lambda' > 0$ as a parameter as well. We will abuse notation and replace λ' with λ, so we have

$$\Pr[f(X_1, \ldots, X_m) \geq (p_0 + \epsilon)n] \leq \left(\prod_{j=1}^{n} \mathbb{E}[\exp(\lambda \cdot f_j(X_{P_j}^*))/\exp(\lambda(p_0 + \epsilon)n]\right)^{1/k},$$

for any $\lambda > 0$. Now, observe that the right hand side of the inequality is the exact same as in the proof of the standard Chernoff bound under independence, except with an additional exponent $1/k$. As a result, we can follow the original proof of the Chernoff bound to show that

$$\Pr[\textstyle\sum_{j=1}^{n} f_j(X_{P_j}) \geq (p_0 + \epsilon)n] \leq \exp(-D(p_0 + \epsilon||p_0) \cdot n/k),$$

which was our desired result. ◀

▶ **Corollary 15.** *Let $X_1, \ldots, X_m$ be 1-negatively associated random variables. Suppose that $f_j(X_{P_j})$ for $j \in [n]$ are a read-k family, where $f_j : \{0,1\}^{P_j} \to [0,1]$ are submodular functions. If we let $p_0 = (1/n)\sum_{j=1}^{n} \mathbb{E}[f_j(X_{P_j}^*)]$ denote the averaged expectation when the underlying random variables are independent, we have*

$$\Pr[\textstyle\sum_{j=1}^{n} f_j(X_{P_j}) \leq (p_0 - \epsilon)n] \leq \exp(-D(p_0 - \epsilon||p_0) \cdot n/k).$$

Proof. Define $g_j := 1 - f_j$, where $g_j : \{0,1\}^{P_j} \to [0,1]$ are supermodular. Then $1 - p_0 = (1/n)\sum_{j=1}^{n} \mathbb{E}[g_j(X^*_{P_j})]$. Applying Theorem 14, we have:

$$\Pr[\textstyle\sum_{j=1}^{n} g_j(X_{P_j}) \geq (1-(p_0-\epsilon))n] \leq (1-p_0+\epsilon)n] \leq \exp(-D(1-p_0+\epsilon||1-p_0)\cdot n/k).$$

The property of Kullback–Leibler divergence $D(1-p||1-q) = D(p||q)$ implies

$$\begin{aligned}\Pr[\textstyle\sum_{j=1}^{n} f_j(X_{P_j}) \leq (p_0-\epsilon)n] &= \Pr[\textstyle\sum_{j=1}^{n} g_j(X_{P_j}) \geq (1-(p_0-\epsilon))n] \\ &\leq \exp(-D(p_0-\epsilon||p_0)\cdot n/k),\end{aligned}$$

which concludes the proof. ◀

References

1. Zeyuan Allen-Zhu and Lorenzo Orecchia. Nearly-linear time positive lp solver with faster convergence rate. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 229–236. Association for Computing Machinery, 2015. doi:10.1145/2746539.2746573.
2. Noga Alon and Joel H. Spencer. *The Probabilistic Method, Third Edition*. Wiley-Interscience series in discrete mathematics and optimization. Wiley, 2008.
3. Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19(3):357–367, 1967. doi:10.2748/tmj/1178243286.
4. Julius Borcea, Petter Brändén, and Thomas M. Liggett. Negative dependence and the geometry of polynomials. *Journal of the American Mathematical Society*, 22(2):521–567, 2009. URL: http://www.jstor.org/stable/40587241.
5. Chandra Chekuri and Kent Quanrud. Submodular function maximization in parallel via the multilinear relaxation. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 303–322. SIAM, 2019. doi:10.1137/1.9781611975482.20.
6. Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010*, pages 575–584. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.60.
7. Herman Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the sum of Observations. *The Annals of Mathematical Statistics*, 23(4):493–507, 1952. doi:10.1214/aoms/1177729330.
8. Devdatt Dubhashi and Desh Ranjan. Balls and bins: a study in negative dependence. *Random Struct. Algorithms*, 13(2):99–124, September 1998. doi:10.1002/(SICI)1098-2418(199809)13:2\%3C99::AID-RSA1\%3E3.0.CO;2-M.
9. Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009. URL: http://www.cambridge.org/gb/knowledge/isbn/item2327542/.
10. Tomás Feder and Milena Mihail. Balanced matroids. In S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis, editors, *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, STOC 1992*, pages 26–38. ACM, 1992. doi:10.1145/129712.129716.
11. Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *J. ACM*, 53(3):324–360, 2006. doi:10.1145/1147954.1147956.
12. Kevin Garbe and Jan Vondrák. Concentration of lipschitz functions of negatively dependent variables, 2018. arXiv:1804.10084.

13 Dmitry Gavinsky, Shachar Lovett, Michael E. Saks, and Srikanth Srinivasan. A tail bound for read-k families of functions. *Random Struct. Algorithms*, 47(1):99–108, 2015. doi:10.1002/rsa.20532.

14 Nicholas J. A. Harvey and Neil Olver. Pipage rounding, pessimistic estimators and matrix concentration. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*, pages 926–945. SIAM, 2014. doi:10.1137/1.9781611973402.69.

15 Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. URL: http://www.jstor.org/stable/2282952.

16 Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan. A (slightly) improved approximation algorithm for metric TSP. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 32–45. ACM, 2021. doi:10.1145/3406325.3451009.

17 Jeong Han Kim and Van H. Vu. Concentration of multivariate polynomials and its applications. *Comb.*, 20(3):417–434, 2000. doi:10.1007/s004930070014.

18 Joseph (Seffi) Naor, Aravind Srinivasan, and David Wajc. Online dependent rounding schemes. *arXiv preprint*, 2024. arXiv:2301.08680.

19 Charles M. Newman. Asymptotic independence and limit theorems for positively and negatively dependent random variables. *Lecture Notes-Monograph Series*, 5:127–140, 1984. URL: http://www.jstor.org/stable/4355492.

20 Alessandro Panconesi and Aravind Srinivasan. Randomized distributed edge coloring via an extension of the chernoff-hoeffding bounds. *SIAM J. Comput.*, 26(2):350–368, 1997. doi:10.1137/S0097539793250767.

21 Robin Pemantle and Yuval Peres. Concentration of lipschitz functionals of determinantal and other strong rayleigh measures. *Comb. Probab. Comput.*, 23(1):140–160, 2014. doi:10.1017/S0963548313000345.

22 Yuval Peres, Mohit Singh, and Nisheeth K. Vishnoi. Random walks in polytopes and negative dependence. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, volume 67 of *LIPIcs*, pages 50:1–50:10. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ITCS.2017.50.

23 Frederick Qiu and Sahil Singla. Submodular dominance and applications. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2022*, volume 245 of *LIPIcs*, pages 44:1–44:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.APPROX/RANDOM.2022.44.

24 Jeanette P. Schmidt, Alan Siegel, and Aravind Srinivasan. Chernoff-hoeffding bounds for applications with limited independence. *SIAM J. Discret. Math.*, 8(2):223–250, 1995. doi:10.1137/S089548019223872X.

25 Maciej Skorski. Tight chernoff-like bounds under limited independence. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2022*, volume 245 of *LIPIcs*, pages 15:1–15:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.APPROX/RANDOM.2022.15.

26 Aravind Srinivasan. Distributions on level-sets with applications to approximation algorithms. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001*, pages 588–597. IEEE Computer Society, 2001. doi:10.1109/SFCS.2001.959935.

27 Alan Tsang, Bryan Wilder, Eric Rice, Milind Tambe, and Yair Zick. Group-fairness in influence maximization. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*, pages 5997–6005. ijcai.org, 2019. doi:10.24963/ijcai.2019/831.

28 Rajan Udwani. Multi-objective maximization of monotone submodular functions with cardinality constraint. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Annual Conference on Neural Information Processing Systems, NeurIPS 2018*, pages 9513–9524, 2018. URL: https://proceedings.neurips.cc/paper/2018/hash/7e448ed9dd44e6e22442dac8e21856ae-Abstract.html.

29 David Wajc. Negative association: Definition, properties, and applications. Manuscript, 2017. URL: https://www.cs.cmu.edu/âĹdwajc/notes/Negative%20Association.pdf.

A Concentration of Read-k Families

We will give a new simpler proof of the results of Gavinsky et al. [13] using exponential moments for f_j for independent random variables $X_1, \ldots, X_n$. Our proof will use the following lemma

▶ **Lemma 16.** *For an arbitrary read-k family $F_1, \ldots, F_n$, we have*

$$\mathbb{E}\left[\prod_{j=1}^{n} F_j\right] \leq \left(\prod_{j=1}^{n} \mathbb{E}\left[F_j^k\right]\right)^{1/k}. \tag{11}$$

Using this lemma, if we define

$$F_j = \exp(\lambda \cdot f_j(X_1, \ldots, X_m)),$$

it is easy to see that $F_1, \ldots, F_n$ are a read-k family since $f_1, \ldots, f_n$ are read-k. We will show later that after applying inequality (11) on $F_1, \ldots, F_m$, we can adapt the standard Chernoff bound proof to our case.

Proof. We will prove inequality (11) via induction on the number of independent variables m. For the base case $m = 1$, observe that there are at most k non-constant functions F_j by definition. If there are fewer than k functions F_j, we can also add identity functions without changing the product. As a result, we can without loss of generality assume that there are exactly k functions F_j. The inequality then follows directly by the Generalized Hölder's Inequality.

Now assume we have proven the statement for m independent variables; we will try to prove it for $m + 1$. Again, let $S = S_1$ denote the set of functions F_j which are influenced by X_1. We have

$$\begin{aligned}
\mathbb{E}\left[\prod_{j=1}^{n} F_j\right] &= \mathbb{E}\left[\prod_{j\in S} F_j \cdot \prod_{j\notin S} F_j\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\prod_{j\in S} F_j \cdot \prod_{j\notin S} F_j \middle| X_2, \ldots, X_{m+1}\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\prod_{j\in S} F_j \middle| X_2, \ldots, X_{m+1}\right] \cdot \prod_{j\notin S} F_j\right]
\end{aligned} \tag{12}$$

Here, the first equality is obvious, the second equality follows by the law of total expectation, and the third equality follows since F_j for $j \notin S$ only depends on $X_2, \ldots, X_{m+1}$ and is independent of X_1.

After taking the conditional expectation, observe that $\mathbb{E}[\prod_{j\in S} F_j | X_2, \ldots, X_{m+1}]$ is a random variable which only depends on $X_2, \ldots, X_{m+1}$. In particular, these form a read-k family over m random variables, so we can apply the inductive hypothesis to claim that

$$\mathbb{E}\left[\prod_{j\in S} F_j | X_2, \ldots, X_{m+1}\right] \leq \left(\prod_{j\in S} \mathbb{E}\left[F_j^k | X_2, \ldots, X_{m+1}\right]\right)^{1/k}$$

After combining this with equation (12), we have

$$\mathbb{E}\left[\prod_{j=1}^{n} F_j\right] \leq \mathbb{E}\left[\prod_{j\in S} \mathbb{E}\left[F_j^k | X_2, \ldots, X_{m+1}\right]^{1/k} \cdot \prod_{j\notin S} F_j\right] =: A$$

For $j \notin S$, let $G_j = F_j$ and for $j \in S$, define

$$G_j = \mathbb{E}\left[F_j^k | X_2, \ldots, X_{m+1}\right]^{1/k}$$

so that $A = \mathbb{E}\left[\prod_{j=1}^{n} G_j\right]$. Observe that G_j are again a read-k family on $X_2, \ldots, X_{m+1}$, so we can again apply the induction hypothesis to claim that

$$A = \mathbb{E}\left[\prod_{j=1}^{n} G_j\right] \leq \prod_{j=1}^{n} \mathbb{E}\left[G_j^k\right]^{1/k} = \prod_{j=1}^{n} \mathbb{E}\left[F_j^k\right]^{1/k},$$

where the final equality follows directly by definition of G_j. This completes the proof. ◀

Using the lemma, we can follow the standard Chernoff bound techniques to complete the proof. Let $X = \sum_{j=1}^{n} f_j(X_1, \ldots, X_m)$; we can apply Markov's inequality for any $\lambda < 0$ to obtain

$$\Pr[X \leq qn] = \Pr[\exp(\lambda X) \geq \exp(\lambda qn)] \leq \exp(-\lambda qn) \cdot \mathbb{E}\left[\exp(\lambda X)\right]$$

As mentioned before, we can take $F_j = \exp(\lambda \cdot f_j(X_1, \ldots, X_m))$ and apply inequality (11) to obtain that

$$\exp(\lambda X) = \mathbb{E}\left[\prod_{j=1}^{n} F_j\right] \leq \prod_{j=1}^{n} \mathbb{E}[F_j^k]^{1/k}.$$

Combining this inequality with the previous inequality, we now have that

$$\Pr[X \leq qn] \leq \exp(-\lambda qn) \cdot \prod_{j=1}^{n} \mathbb{E}[F_j^k]^{1/k} = \left(\prod_{j=1}^{n} \mathbb{E}\left[[F_j/\exp(\lambda q)]^k\right]\right)^{1/k}.$$

Writing out the definition of F_j, we have

$$[F_j/\exp(\lambda q)]^k = \exp(\lambda k \cdot f_j(X_1, \ldots, X_m))/\exp(\lambda k q).$$

Define $\lambda' = \lambda k$; since λ was arbitrary, we will abuse notation and let $\lambda = \lambda'$. Combining the two previous inequalities, we have that

$$\Pr[X \leq qn] \leq \left(\prod_{j=1}^{n} \mathbb{E}[\exp(\lambda \cdot f_j)/\exp(\lambda q)]\right)^{1/k}.$$

Here, the right-hand side is in exactly the same form as in the proof of the Chernoff-Hoeffding theorem under independence, except with an additional exponent $1/k$. As a result, we can follow the Chernoff-Hoeffding proof and take $q = p - \epsilon$ to obtain

$$\Pr[X \leq (p-\epsilon)n] \leq \exp(-D(p-\epsilon||p) \cdot n/k),$$

which was our desired result.

Robust Restaking Networks

Naveen Durvasula
Columbia University, New York, NY, USA

Tim Roughgarden
a16zcrypto, New York, NY, USA
Columbia University, New York, NY, USA

Abstract

We study the risks of validator reuse across multiple services in a restaking protocol. We characterize the robust security of a restaking network as a function of the buffer between the costs and profits from attacks. For example, our results imply that if attack costs always exceed attack profits by 10%, then a sudden loss of .1% of the overall stake (e.g., due to a software error) cannot result in the ultimate loss of more than 1.1% of the overall stake. We also provide local analogs of these overcollateralization conditions and robust security guarantees that apply specifically for a target service or coalition of services. All of our bounds on worst-case stake loss are the best possible. Finally, we bound the maximum-possible length of a cascade of attacks.

Our results suggest measures of robustness that could be exposed to the participants in a restaking protocol. We also suggest polynomial-time computable sufficient conditions that can proxy for these measures.

2012 ACM Subject Classification Networks → Network economics

Keywords and phrases Proof of stake, Restaking, Staking Risks

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.48

Related Version *Full Version*: `https://arxiv.org/abs/2407.21785`

Funding This research was supported in part by NSF awards CCF-2006737 and CNS-2212745, and research awards from the Briger Family Digital Finance Lab and the Center for Digital Finance and Technologies.

Acknowledgements We thank Tarun Chitra, Soubhik Deb, Sreeram Kannan, Mike Neuder, and Mallesh Pai for comments on earlier drafts of this paper. We thank Soubhik and Sreeram in particular for emphasizing the importance of local guarantees.

1 Introduction

1.1 Sharing Validators Across Services

Major blockchain protocols such as Bitcoin or Ethereum are "decentralized," meaning that transaction execution is carried out by a large and diverse set of "validators." Such protocols offer a form of "trusted computation," in the sense that, because they are decentralized, no one individual or entity can easily interfere with their execution. A decentralized and Turing-complete smart contract platform such as Ethereum can then be viewed as a trusted programmable computer capable of performing arbitrary computations.

While Turing-complete, the computing functionality offered by Ethereum smart contracts suffers from limitations imposed by design decisions in the underlying consensus protocol. Most obviously, computation and storage in the Ethereum virtual machine is scarce, with perhaps 15–20 transactions processed per second. Could the Ethereum protocol be somehow bypassed, opening the door for different or more powerful computing functionality, while retaining at least some of the protocol's decentralization? Or, what about applications that are not compatible with all Ethereum validators, perhaps due to demanding hardware requirements or regulatory constraints?


© Naveen Durvasula and Tim Roughgarden;
licensed under Creative Commons License CC-BY 4.0



One natural approach to addressing these challenges is to allow the *reuse* of a blockchain protocol's validators across multiple services, where a "service" is some task that could be carried out by some subset of validators. (The initial blockchain protocol can be viewed as the canonical service, performed by all validators.) For example, such services could include alternative consensus protocols (perhaps with higher throughput, a different virtual machine, or different consistency-liveness trade-offs), storage ("data availability"), or verifiable off-chain computation ("zk coprocessors").[1]

The obvious danger of validator reuse is an increased risk of a validator deviating from its intended behavior (e.g., due to overwhelming computational responsibilities). Our focus here is deliberate validator deviations in response to economic incentives, such as the profits that could be obtained by corrupting one or more services. The goal of this paper is to quantify such risks:

Under what conditions can validators be safely reused across multiple services?

1.2 Cryptoeconomic Security

We first review the usual "cryptoeconomic" approach to answering a more basic question: when is a blockchain protocol, without any additional services, "safe from attack"? The idea is to perform a cost-benefit analysis from the perspective of an attacker, and declare the protocol safe if the cost of carrying out an attack exceeds the profit that the attacker can expect from it; see also Figure 1. In the specific case of a proof-of-stake blockchain protocol with slashing (such as Ethereum), the cost can be estimated as the value of the validator stake that would be lost to slashing following an attack. For example, let V denote the set of validators of a proof-of-stake blockchain protocol and σ_v the stake of validator $v \in V$ (e.g., 32 ETH in Ethereum). In a typical PBFT-type protocol in which double-voting validators lose all of their stake, the cost of an attack (i.e., causing a consistency violation) can be bounded below by $\frac{1}{3}\sum_{v \in V} \sigma_v$. In this case, the protocol can be regarded as cryptoeconomically secure provided the estimated profit π of an attack is less than this quantity. To the extent that there is a "buffer" between $\frac{1}{3}\sum_{v \in V} \sigma_v$ and π, the protocol can be treated as "robustly secure," meaning secure even after a sudden loss of some amount of stake (e.g., due to slashing caused by a software error).

1.3 Our Results: Robustly Secure Validator Reuse

Viewing the basic scenario of a blockchain protocol as a star graph (with one "service" representing the protocol connected to all the validators running it), the more complex scenario of validators reused across multiple services can be viewed as an arbitrary bipartite graph (Figure 2). As before, we suppose that each validator $v \in V$ has some stake σ_v that can be confiscated in the event that v participates in an attack on a service that it has agreed to support. There is now a set S of services, with π_s denoting the profit an attacker would obtain by compromising the service $s \in S$.[2] We assume that compromising a service $s \in S$ requires the participation of an α_s fraction of the overall validator stake supporting v (e.g., $\alpha_s = 1/3$). With this expanded network formalism to capture multiple services, when should we consider a network to be "secure"?

[1] Several projects, in various stages of production, are currently exploring this idea; the Eigenlayer restaking protocol [15] is perhaps the most well known of them. We stress that our goal here is to develop a model that isolates some of the fundamental challenges and risks of validator reuse, independent of any specific implementation of the idea.

[2] We follow [15] and assume that the estimates on attack profitability (the π_s's) are given. Developing tools to help produce such estimates in practice is an important open research direction.

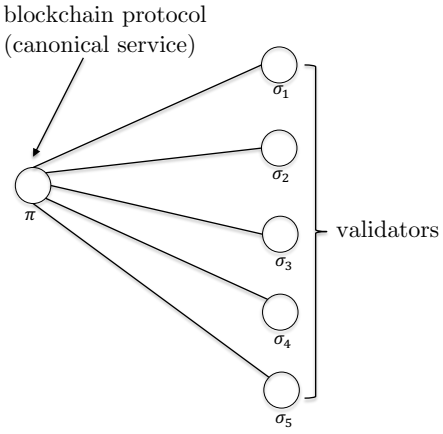


Figure 1 A blockchain protocol operated by a collection of validators, with π denoting the profit of successfully attacking the protocol and σ_v the amount of stake posted by a validator v as collateral.

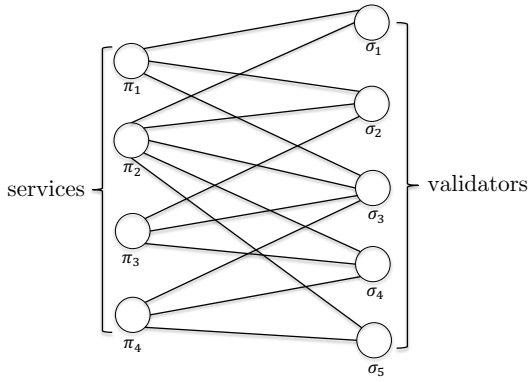
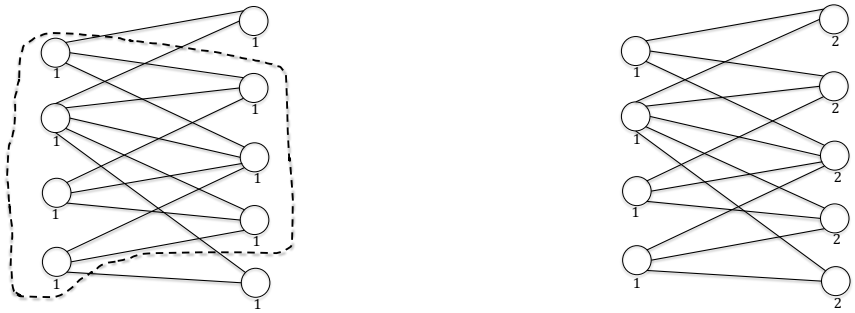


Figure 2 A general restaking network, with validators reused across multiple services.

Intuitively, a network is insecure if there is a set of services that can be profitably attacked. This requires two conditions to be satisfied: the validators $B \subseteq V$ carrying out the attack must control sufficient stake to do so (for each service s in the attacked set A, the validators of B control at least a α_s fraction of the staked pledged to s), and they must profit from it (i.e., $\sum_{s \in A} \pi_s > \sum_{v \in B} \sigma_v$). We call such a pair (A, B) a *valid attack* and a network *secure* if there are no valid attacks; see also Figure 3. With multiple services, security is an inherently combinatorial (as opposed to binary) notion. For example, the computational problem of checking whether a network is secure is as hard as the (coNP-hard) problem of verifying the expansion of a bipartite graph (see [12]).

When is a network "robustly secure," in the sense that the sudden loss of a small amount of stake cannot enable a catastrophic attack? Unlike the "all-or-nothing" version of this question with a single service, with multiple services, the following more fine-grained question is the appropriate one: given an initial shock in the form of a sudden loss of a ψ fraction of the overall stake, what is the total fraction of stake that might be lost following any consequent valid attacks?

As in the case of a single service, some amount of "buffer" in stake (relative to attack profits) is necessary for robust security. We parameterize this overcollateralization factor via a parameter γ and suppose that, whenever $B \subseteq V$ is a subset of validators capable of



(a) A valid attack. **(b)** A secure network.

Figure 3 Two restaking networks. Each service (left-hand side vertex) and validator (right-hand side vertex) is labeled with its profit-from corruption or stake, respectively. Assume that a service can be corrupted if and only if it is attacked by at least half of its validators (i.e., $\alpha_s = 1/2$ for every service s). The restaking network in (a) is not secure because there is a valid attack (indicated by the dotted line): three validators can earn a profit of 4 by corrupting all four services while losing only three units of stake. The restaking network in (b) is secure.

corrupting all the services in $A \subseteq S$, the total stake of B is at least a $1+\gamma$ factor larger than the total profit from corrupting all of A. For example, this condition holds in the network in Figure 3(b) for $\gamma = 1/2$ (but not for larger values of γ).

Our first main result (Theorem 4) precisely characterizes the worst-case (over bipartite graphs and shocks, as a function of γ) fraction of the overall stake that can be lost due to a shock of size ψ: $\left(1+\frac{1}{\gamma}\right)\psi$. Because the network was secure prior to the shock, the value of a $\left(1+\frac{1}{\gamma}\right)\psi$ fraction of the overall stake is also an upper bound on the total profit obtained from all of the attacked services.

We also show that our result is tight in a strong sense (Theorem 6 and Theorem 7): for every ψ, γ, and ϵ greater than zero such that $0 \leq \left(1+\frac{1}{\gamma}\right)\psi - \epsilon \leq 1$, there exists a restaking graph in which a ψ fraction of the overall stake can disappear in a shock that results in the loss of at least a $\left(1+\frac{1}{\gamma}\right)\psi - \epsilon$ fraction of the overall stake.[3]

Qualitatively, this result implies that a constant-factor strengthening of the obvious necessary condition for security automatically implies robust security. For example, if attack costs always exceed attack profits by 10%, then a sudden loss of .1% of the overall stake cannot result in the ultimate loss of more than 1.1% of the overall stake.

Our result suggests a "risk measure" that could be exposed to the participants in a restaking protocol, namely the maximum value of the buffer parameter γ that holds with respect to the current restaking network. We also suggest easy-to-check sufficient conditions that can proxy for this risk measure (Corollary 5). These conditions are similarly tight, as shown in Theorem 6 and Theorem 7.

1.4 Our Results: Local Robust Security Guarantees

The results described in Section 1.3 are "global" with respect to the network structure, in three distinct senses: (i) the overcollateralization condition is assumed to hold for every subset $B \subseteq V$ of validators and $A \subseteq S$ of services that B is capable of corrupting; (ii) the

[3] After slightly reducing the validator stakes, the network in Figure 3(b) already shows that the bound is tight for the special case in which $\psi = 1/5$ and γ is arbitrarily close to $1/2$. (Consider a shock that knocks out the validator that is connected to all four services.)

initial shock can affect any subset of validators, subject to the assumed bound of ψ on the total fraction of stake lost; and (iii) any subset of validators might lose stake following the initial shock, subject to our upper bound of $(1 + \frac{1}{\gamma})\psi$ on the total fraction of lost stake.

Local guarantees

We next pursue more general "local" guarantees, which are parameterized by a set C of services. (The global guarantees will correspond to the special case in which $C = S$.) For example, C might be a set of closely related services that share a number of dedicated validators. The operators of such a set C might object to both the assumptions and the conclusion of our global guarantee:

- How can we be sure that the overcollateralization factor holds for services and validators that we know nothing about?
- And even if we could, how can we be sure that random validators that we have nothing to do with won't suddenly lose their stake (e.g., because they supported a malicious or buggy service), resulting in an initial shock the causes the loss of more than a ψ fraction of the overall stake?
- And even if we could, how can we be sure that our validators won't be the ones that lose their stake following a shock that is purely the fault of other services and/or validators?

To address these concerns, we next consider a refined version of the basic model, parameterized by a set C of services. We denote by $\Gamma(C)$ the validators that are exclusive to C, meaning that they contribute to no services outside C. (In the special case in which $C = S$, $\Gamma(C) = V$ and we recover the original model.) Intuitively, the validators in $\Gamma(C)$ are the ones that services in C are "counting on"; other validators support (potentially malicious or buggy) services outside C and, from C's perspective, could disappear at any time. We then restrict attention to initial shocks in which at most a ψ fraction of the total stake controlled by the validators of $\Gamma(C)$ is lost. (The shock can affect validators outside of $\Gamma(C)$ arbitrarily.) The goal is then to identify overcollateralization conditions guaranteeing that, no matter what the initial shock and subsequent attacks, the fraction of stake ultimately lost by the validators in $\Gamma(C)$ is bounded (by a function of the shock size ψ and an overcollateralization parameter γ).

Generalizing our global guarantees to local guarantees is not straightforward and, as the next two examples show, requires additional compromises. The first example shows that protection can be guaranteed only against a subset of valid attacks, a natural and well motivated subset that we call "stable attacks." The second example shows that, even when restricting attention to stable attacks, overcollateralization is required not only for potential attacks, but more generally for what we call "attack headers."

Necessity of restricting to stable attacks

In more detail, in the first example (depicted on the left of Figure 4), we consider a restaking network with two services and three validators. (Vertices in this figure correspond to validators, which were previously represented as the right-hand side vertices of a bipartite graph; each service is now identified with its neighborhood of validators.) The service highlighted in green on the left has 100 times as much stake securing it as its profit from corruption, and shares no validators with other services, so one might hope that it would be protected from any shocks that affect only the other validators. However, if the validator outlined in red disappears, the two validators outlined in yellow can attack both services for a profit of $102 - 101 > 0$. But this example is unsatisfying: the validator with stake 1

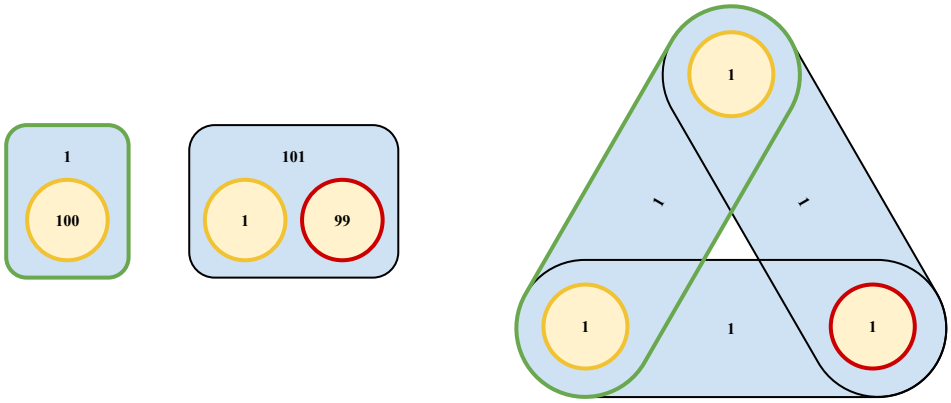


Figure 4 Simple overcollateralization is insufficient in the local setting. There are two restaking networks shown above. In each, the validators are denoted along with their corresponding stakes by the yellow circles. Services and their profits from corruption are denoted by the blue rounded squares. In each of these networks, the service outlined in green is overcollateralized. However, despite being unrelated to the overcollateralized service, if the validator outlined in red disappears, the validators in yellow can attack all services (including the overcollateralized one).

could have attacked the service with profit from corruption 101 on its own to yield a profit of 100. The addition of the validator with stake 100 added 100 to the cost of the attack, but only added 1 (from the corruption of the service highlighted in green) to the profit. We show in Theorem 8 that the issue suggested by this example is fundamental: without further restrictions on attacks that rule out contrived examples such as this one, there does not exist a local condition that guarantees local security. In response, we confine attention to sequences of what we call *stable* attacks in which all of the attacking validators contribute positively to the profit of the attack (as opposed to free riding on the profits attributable to other attacking validators). The attack in the example is not stable, as the validator with stake 100 was not a profitable addition to the attack that could have been carried out by the validator with stake 1.

Necessity of overcollateralizing attack headers

Even if we restrict our attention to stable attacks, simple overcollateralization is insufficient to guarantee local security. To appreciate the issue, consider the restaking network shown on the right in Figure 4. In this example, there are three services, each with a profit from corruption of 1, and three services each with stake 1. Each validator is used to secure two different services. The service outlined in green is overcollateralized in that its profit from corruption is 1, but two units of stake stake are securing it. Despite this, if the unrelated validator outlined in red disappears, the two validators outlined in yellow can attack all three services for a profit of $3 - 2 > 0$. Furthermore, if all validators are required to attack each service (i.e., $\alpha_s = 1$), then this attack is stable, as the inclusion of both validators in the attack is profitable. Thus, despite being overcollateralized, a stable attack can be launched on the service highlighted in green even if stake unrelated to the service disappears in a shock.

We find that to guarantee robust security for a coalition of services C using only "local" information, it is necessary to overcollateralize not only pairs (A, B) where B is a set of validators capable of attacking all services in A, but a more general collection of pairs that we

call *attack headers*. Informally, this amounts to requiring that there is some "buffer" in stake for any potential attack on some services in C even if we were to allow every validator that is also securing a service outside of C to join the attack without considering their profitability. Our main result here (Theorem 10) formally provides a local condition guaranteeing that, whenever an initial shock knocks out at most a ψ fraction of the stake that provides security exclusively to C, the worst-case loss of such stake, after an arbitrary sequence of stable attacks, is at most a $(1 + \frac{1}{\gamma})\psi$ fraction. We show that our bounds are tight and indeed require overcollateralization of all attack headers (Corollary 14 and Theorem 15), and again provide easily computable sufficient (and similarly tight) conditions that can proxy for the overcollateralization condition (Corollary 13). Our local condition generalizes the global overcollateralization condition, with the latter corresponding to the special case of the former in which $C = S$.

1.5 Our Results: Cascading Attacks

An attack on a restaking network results in a loss of stake (of the attacking validators), and this may introduce new opportunities for other sets of validators to carry out profitable attacks. That is, an initial shock may set off an entire *cascade* of attacks. (All of the bounds on stake loss described in Sections 1.3 and 1.4 hold for cascades of attacks of arbitrary length.) Our final result concerns the maximum-possible length of such a cascade, and shows that this quantity is also governed in part by the overcollateralization factor γ.

Precisely, we define the *reference depth* of a cascade of attacks as a measure of the "long-range dependence" between different attacks in an attack sequence. For example, if each attack in the sequence is directly enabled by the loss of the validators slashed in the previous attack, then the reference depth of the sequence is 1. Our main result here (Theorem 16) bounds the maximum-possible attack length as a function of the reference depth k, the shock size ψ, the overcollateralization factor γ, and the minimum stake ϵ held by a validator: $k(1 + \log_{1+\gamma}(\frac{\psi \cdot [\text{total stake}]}{\epsilon\gamma}))$. For example, in the case of constant reference depth and equal validator stake amounts, the worst-case attack length is logarithmic in the number of validators with overcollateralization and linear without it.

1.6 Related Work

Our focus on the risks of cascading failures following a small shock echoes some of themes in the literature on systemic risk in financial networks. For example, Eisenberg and Noe [9] that study the existence and structure of inter-firm payments in a financial network following a default. This work is extended by Glasserman and Young [11], who study how "connectedness," meaning the fraction of liabilities that a firm externally owes, affects contagion risk. Acemoglu et al. [2] build further on this work and study network "stability," meaning the propensity for shocks to propagate; they show that connectivity initially improves stability, but then at a phase transition, denser connectivity leads to increased shock propagation. In a subsequent paper, Acemoglu et al. [1] unify a number of the preceding results.

A separate line of work, beginning with Chen et al. [6], aims to axiomatically characterize systemic risk measures. The model in [6] can capture, in particular, a contagion model characterized by a matrix of profits and losses over different firms and outcomes. The results in [6] characterize the global measures of risk (operating on the matrix) that satisfies certain sets of desirable axioms. This is work is expanded upon in Kromer et al. [13], where the authors consider general outcome measure spaces, as well as in Feinstein et al. [10], where the authors consider set-valued risk measures. Battison et al.[4] study systemic risk measurement

when a regulator has limited information about contracts made between financial network participants (e.g., the fraction of the face value that can likely be recovered if a counterparty defaults), and show how small errors in knowledge can lead to large errors in systemic risk measurement.

Our work also shares some conceptual similarity with the well-known work of Diamond and Dybvig [8] on bank runs and of Brunnermeier et al. [5] on the risks of re-hypothecation.

The model in the present work differs substantially from those considered in the aforementioned papers, in large part because of the idiosyncrasies of restaking networks, including their combinatorial and bipartite nature and their susceptibility to economically motivated attacks.

Restaking has also, to a limited extent, been studied in its own right. The EigenLayer team introduces restaking in [15]. The framing of an economically motivated attacker that trades off stake loss with profits from corruption appears in [15], and is used also by Deb et al. [7]. Chitra and Neuder [14] discuss restaking risk from a validator perspective, comparing restaking with investments in other financial instruments (e.g. bonds). Alexander [3] considers the interplay between existing leveraging schemes and liquid restaking tokens, which can amplify large-scale credit risk.

2 Model

Validators and Services

We consider a setting in which there is a set V of validators and a set S of services. Each service $s \in S$ has some profit from corruption π_s, and each validator $v \in V$ has some stake σ_v. We also associate with each service s a parameter α_s that denotes the fraction of stake required to corrupt/launch an attack on s. We call a bipartite graph $G = (S, V, E, \pi, \sigma, \alpha)$ a *restaking graph*; an edge is drawn between a validator $v \in V$ and a service $s \in S$ if v is restaking for s. For a given set of vertices A in a graph G, we use the notation $N_G(A)$ to denote the neighbors of A.

Attack Dynamics

For simplicity, we assume that validators lose their full stake σ_v if they launch an attack on a service. As such, for a given collection of services $A \subseteq S$, and a given collection of validators $B \subseteq V$ restaking for those services, we say that (A, B) is an *attacking coalition* for a restaking graph G if the validators in B possess enough stake to corrupt the services A:

$$\underbrace{\sum_{v \in B \cap N_G(\{s\})} \sigma_v}_{\text{Total stake in } s \text{ owned by validators } B} \geq \alpha_s \cdot \underbrace{\sum_{v \in N_G(\{s\})} \sigma_v}_{\text{Total amount restaked in } s} \qquad \forall s \in A \tag{1}$$

We further say that (A, B) is a *valid attack* if it is an attacking coalition that has an incentive to launch an attack:

$$\underbrace{\sum_{s \in A} \pi_s}_{\text{Total profit from corrupting } A} > \underbrace{\sum_{v \in B} \sigma_v}_{\text{Total stake owned by validators } B} \tag{2}$$

If a valid attack (A, B) is carried out, we denote by $G \searrow B$ the induced subgraph $G[S, V \setminus B]$. The graph $G \searrow B$ denotes the state of the restaking graph after the attack is carried out. If no valid attacks exist on the graph, we call it *secure*. To simplify notation, for any subset of validators $B \subseteq V$, we will use the shorthand σ_B to denote $\sum_{v \in B} \sigma_v$. Similarly, for any $A \subseteq S$, we will use π_A as shorthand for $\sum_{s \in A} \pi_s$.

EigenLayer sufficient conditions

We note in passing that, in their whitepaper [15], EigenLayer proposes some efficiently verifiable sufficient conditions for security that they check to ensure that an attack does not exist.

▷ Claim 1 (EigenLayer Sufficient Conditions, from Appendix B.1 of the EigenLayer Whitepaper [15]). A restaking graph G is secure if for each validator $v \in V$,

$$\sum_{s \in N_G\{v\}} \frac{\sigma_v}{\sigma_{N_G(\{s\})}} \cdot \frac{\pi_s}{\alpha_s} \leq \sigma_v \tag{3}$$

Proof. It is shown in Appendix B.1 of the whitepaper that if Equation (3) holds for G, then the graph is secure (i.e. no valid attacks (A, B) satisfying Equations (1) and (2) exist). ◁

Cascading attacks

Our goal is to understand when a small shock can result in the loss of a large fraction of the overall stake. Small shocks can turn into large shocks by means of a cascading attack. Formally, we say that a disjoint sequence $(A_1, B_1), \ldots, (A_T, B_T) \in 2^S \times 2^V$ is a *valid cascade of attacks* on a restaking graph $G = (S, V, E, \pi, \sigma, \alpha)$ if for each $t \in [T]$, (A_t, B_t) is a valid attack on $G \searrow \bigcup_{i=1}^{t-1} B_i$. We denote by $\mathcal{C}(G)$ the set of all such sequences of valid cascading attacks.

Worst-case stake loss

We now define a metric that measures the total potential loss of stake due to a sequence of cascading attacks. In our model, we first suppose that an initial small shock decreases the amount of stake. Formally, we define, for a given restaking graph $G = (S, V, E, \pi, \sigma, \alpha)$,

$$\mathbb{D}_\psi(G) := \left\{ D \subseteq V \mid \frac{\sigma_D}{\sigma_V} \leq \psi \right\} \tag{4}$$

to be the set of validator coalitions that constitute at most a ψ-fraction of all stake. Given some $D \in \mathbb{D}_\psi(G)$, we use the notation $G \searrow D := G[S, V \setminus D]$ to denote the induced subgraph of the restaking graph when we delete the validators D. We now define

$$R_\psi(G) := \underbrace{\psi}_{\text{Initial shock}} + \max_{D \in \mathbb{D}_\psi(G)} \max_{(A_1,B_1),\ldots,(A_T,B_T) \in \mathcal{C}(G \searrow D)} \underbrace{\frac{\sigma_{\bigcup_{t=1}^T B_t}}{\sigma_V}}_{\text{Stake lost from cascading attacks}} \tag{5}$$

This quantity represents the worst-case total fraction of stake lost due to an initial ψ-fraction of the stake disappearing. By construction, $\psi \leq R_\psi(G) \leq 1$.

3 Overcollateralization Provides Robust Security

In this section, we show that "scaling up" the definition of security automatically results in robust security, meaning bounded losses from cascading attacks that follow an initial shock. We first show that, without loss of generality, it suffices to consider single valid attacks $(A, B) \in \mathcal{C}(G \searrow D)$ instead of more general cascading attacks.

▶ **Lemma 2.** *Let $G = (S, V, E, \pi, \sigma, \alpha)$ be an arbitrary restaking graph, and further suppose that (A, B) is an attacking coalition on $G \searrow D$, where $D \subseteq V$. Then, $(A, B \cup D)$ is an attacking coalition on G.*

Proof. Because (A, B) is an attacking coalition on $G \searrow D$, we must have by Equation (1) that

$$\sigma_{B \cap N_G\{s\}} \geq \alpha_s \cdot \sigma_{N_G\{s\} \setminus D} \qquad \forall s \in A \tag{6}$$

It follows that for any $s \in A$,

$$\begin{aligned} \sigma_{(B \cup D) \cap N_G\{s\}} &= \sigma_{B \cap N_G\{s\}} + \sigma_{D \cap N_G\{s\}} && (7)\\ &\geq \alpha_s \cdot \sigma_{N_G\{s\} \setminus D} + \sigma_{D \cap N_G\{s\}} && (8)\\ &\geq \alpha_s \cdot \sigma_{N_G\{s\}} && (9) \end{aligned}$$

and the desired result follows. ◀

▶ **Corollary 3.** *Let $G = (S, V, E, \pi, \sigma, \alpha)$ be an arbitrary restaking graph, and further suppose that $(A_1, B_1), \ldots, (A_T, B_T) \in \mathcal{C}(G)$ is a valid sequence of cascading attacks on G. Then, $\left(\bigcup_{t=1}^T A_t, \bigcup_{t=1}^T B_t\right)$ is also a valid attack on G.*

Proof. By repeatedly applying Lemma 2, we find that for each $t \in [T]$, $\left(A_t, \bigcup_{i=1}^t B_t\right)$ is an attacking coalition on G. It follows by inspection of Equation (1) that we must therefore have that $(\bigcup_t A_t, \bigcup_t B_t)$ is an attacking coalition on G. To finish the result, it suffices to show that Equation (2) holds for this attacking coalition on the original graph G. This follows from the disjointness of the A_t's and of the B_t's:

$$\pi_{\bigcup_t A_t} = \sum_{t=1} \pi_{A_t} > \sum_{t=1}^{T} \sigma_{B_t} = \sigma_{\bigcup_t B_t} \tag{10}$$

where in the inner inequality we use that for each $t \in [T]$, $\pi_{A_t} > \sigma_{B_t}$ by Equation (2), as (A_t, B_t) is a valid attack on $G \searrow \bigcup_{i=1}^{t-1} B_i$. It follows that $(\bigcup_t A_t, \bigcup_t B_t)$ is a valid attack on G. ◀

Adding Multiplicative Slack

Our condition is given by adding multiplicative slack to Equation (2). Formally, we say that a restaking graph G is *secure with γ-slack* if for all attacking coalitions (A, B) on G,

$$(1+\gamma) \underbrace{\pi_A}_{\text{Total profit from corrupting } A} \leq \underbrace{\sigma_B}_{\text{Total stake owned by validators } B} \tag{11}$$

▶ **Theorem 4.** *Suppose that a restaking graph $G = (S, V, E, \pi, \sigma, \alpha)$ is secure with γ-slack for some $\gamma > 0$. Then, for any $\psi > 0$, $R_\psi(G) < \left(1 + \frac{1}{\gamma}\right)\psi$.*

Proof. Take any $\psi > 0$ and any $D \in \mathbb{D}_\psi(G)$ for some restaking graph G where (11) holds. Let $(A_1, B_1) \ldots, (A_T, B_T) \in \mathcal{C}(G \searrow D)$ be arbitrary. Applying Corollary 3, we must have that $(\bigcup_t A_t, \bigcup_t B_t) \in \mathcal{C}(G \searrow D)$ as well. Defining $A := \bigcup_t A_t$ and $B := \bigcup_t B_t$, we must therefore have that (A, B) is an attacking coalition on $G \searrow D$, and furthermore that

$$\pi_A > \sigma_B \tag{12}$$

By Lemma 2, we must also have that $(A, B \cup D)$ is an attacking coalition on the original graph G. It then follows that as G is secure with γ-slack, Equation (11) must hold on $(A, B \cup D)$, whence

$$(1+\gamma)\pi_A \leq \sigma_{B \cup D} = \sigma_B + \sigma_D \tag{13}$$

Putting this together with Equation (12), we find that

$$(1+\gamma)\sigma_B < (1+\gamma)\pi_A \leq \sigma_B + \sigma_D \leq \sigma_B + \psi \cdot \sigma_V \tag{14}$$

It follows that

$$\gamma \cdot \sigma_B < \psi \cdot \sigma_V \implies \frac{\sigma_B}{\sigma_V} < \frac{\psi}{\gamma} \tag{15}$$

$$\implies \psi + \frac{\sigma_B}{\sigma_V} < \left(1 + \frac{1}{\gamma}\right)\psi \tag{16}$$

As we took $A_1, \ldots, A_T$ to be arbitrary, we find that $R_\psi(G) < \left(1 + \frac{1}{\gamma}\right)\psi$, as desired. ◀

The EigenLayer sufficient conditions (3) can be similarly "scaled up" to yield efficiently checkable sufficient conditions for security with γ-slack (and hence, by Theorem 4, robustness to cascading attacks).

▶ **Corollary 5.** *Let G be a restaking graph such that, for all validators $v \in V$,*

$$\sum_{s \in N_G(\{v\})} \frac{\sigma_v}{\sigma_{N_G(\{s\})}} \cdot \frac{(1+\gamma)\pi_s}{\alpha_s} \leq \sigma_v \tag{17}$$

Then, $R_\psi(G) < \left(1 + \frac{1}{\gamma}\right)\psi$.

Proof. This follows from Theorem 4 and Claim 1. Noting that the γ-slack condition holds precisely iff no valid attacks exist when profits from corruption π_s are inflated by a multiplicative factor of $(1+\gamma)$, it suffices to apply EigenLayer's sufficient conditions from Claim 1 with modified profits from corruption $(1+\gamma)\pi_s$. ◀

Note that, given a restaking network, it is straightforward to compute the minimum value of γ such that the condition in (17) holds. This value can then be interpreted as an easily computed "risk measure" of such a network.

4 Lower Bounds for Global Security

In this section, we show that the upper bounds from the previous section are tight. We first show that if there is no multiplicative slack (i.e. $\gamma = 0$), then very small shocks can cause all stake to be lost in the worst case. This holds even under the EigenLayer conditions (3)[4].

▶ **Theorem 6.** *For any $0 < \epsilon < 1$, there exists a restaking graph G that is secure and meets the EigenLayer condition (3), but has $R_\psi(G) = 1$ for all $\psi \geq \epsilon$.*

Proof. We construct a restaking graph $G = (S, V, E, \pi, \sigma, \alpha)$ with one service $S = \{x\}$ and two validators $V = \{a, b\}$, where an edge exists between each validator and the service (i.e. $E := (\{x\}, \{a\}), (\{x\}, \{b\})$). We then let $\sigma_a := \epsilon$, $\sigma_b := 1 - \epsilon$, $\pi_x := 1$, and $\alpha_x := 1$. Without loss of generality, we may assume $\psi < 1$ as $R_1(G) = 1$ for any restaking graph G. This graph satisfies (3) as

[4] The graph exhibited in the proof of Theorem 6 has $\pi_x/\sigma_a \to \infty$ as $\epsilon \to 0$. It is possible to construct a counterexample with similar properties while maintaining that π_s/σ_v is greater than some universal constant for any $s \in S$ and $v \in V$. This is done in the full version of the paper.

$$\frac{\sigma_a}{\sigma_a+\sigma_b}\cdot\frac{\pi_x}{\alpha_x}=\sigma_a \tag{18}$$

$$\frac{\sigma_b}{\sigma_a+\sigma_b}\cdot\frac{\pi_x}{\alpha_x}=\sigma_b \tag{19}$$

whence the graph is also secure by Claim 1. We now consider an initial shock $D=\{a\}$. As $\sigma_a/\sigma_V=\epsilon\leq\psi$, it follows that $D\in D_\psi(G)$. The pair $(\{x\},\{b\})$ is a valid attack on $G\searrow D$, since $\{b\}=N_{G\searrow D}\{x\}$ whence it is an attacking coalition, and $\sigma_b<\pi_x$. It follows that $R_\psi(G)\geq\frac{\sigma_a+\sigma_b}{\sigma_V}=1$ as desired. ◀

Next, we show that the bound we give in Theorem 4 (indeed, more strongly, the condition given in Corollary 5) is tight for all $\psi,\gamma>0$.

▶ **Theorem 7.** *For any $\psi,\gamma,\epsilon>0$ such that*

$$0\leq\left(1+\frac{1}{\gamma}\right)\psi-\epsilon\leq 1, \tag{20}$$

there exists a restaking graph G that satisfies the condition (17) from Corollary 5 but has $R_\psi(G)\geq\left(1+\frac{1}{\gamma}\right)\psi-\epsilon$.

Proof. We construct a restaking graph $G=(S,V,E,\pi,\sigma,\alpha)$ with three validators $V=\{a,b,c\}$ and one service $S=\{x\}$, where an edge exists between each of the validators a and b and the service x (i.e. the edge set $E:=\{(x,a),(x,b)\}$). Without loss of generality, suppose that $\epsilon\leq\psi/\gamma$. Let $\sigma_a>0$ be any positive constant. We define

$$\sigma_b:=\sigma_a\left(\frac{1}{\gamma}-\frac{\epsilon}{\psi}\right) \tag{21}$$

$$\sigma_c:=\sigma_a\left(\frac{1-\psi+\epsilon}{\psi}-\frac{1}{\gamma}\right) \tag{22}$$

$$\pi_x:=\frac{\left(1+\frac{1}{\gamma}\right)\psi-\epsilon}{1+\gamma}\cdot\sigma_V \tag{23}$$

$$\alpha_s:=1 \tag{24}$$

Notice first that $\sigma_b\geq 0$ as we have taken $\epsilon\leq\psi/\gamma$. Next, observe that

$$\sigma_c\geq 0\iff\frac{1-\psi+\epsilon}{\psi}\geq\frac{1}{\gamma} \tag{25}$$

$$\iff 1\geq\left(1+\frac{1}{\gamma}\right)\psi-\epsilon \tag{26}$$

whence $\sigma_c\geq 0$ by Equation (20). Finally, we must also have that $\pi_x\geq 0$ by Equation (20) as well. Next, notice by construction that

$$\sigma_V=\sigma_a+\sigma_b+\sigma_c=\sigma_a\left(1+\frac{1}{\gamma}-\frac{\epsilon}{\psi}+\frac{1-\psi+\epsilon}{\psi}-\frac{1}{\gamma}\right)=\frac{\sigma_a}{\psi} \tag{27}$$

This graph meets condition (17) from Corollary 5, as

$$\sum_{s\in N_G\{a\}}\frac{\sigma_a}{\sigma_{N_G\{s\}}}(1+\gamma)\pi_s=\frac{\sigma_a}{\sigma_a+\sigma_b}\left[\left(1+\frac{1}{\gamma}\right)\psi-\epsilon\right]\sigma_V \tag{28}$$

$$=\frac{\sigma_a}{\sigma_a\left(1+\frac{1}{\gamma}-\frac{\epsilon}{\psi}\right)}\left[\left(1+\frac{1}{\gamma}\right)\psi-\epsilon\right]\frac{\sigma_a}{\psi} \tag{29}$$

$$=\sigma_a \tag{30}$$

A similar argument shows that

$$\sum_{s \in N_G\{b\}} \frac{\sigma_b}{\sigma_{N_G\{s\}}}(1+\gamma)\pi_s = \frac{\sigma_b}{\sigma_a\left[\left(1+\frac{1}{\gamma}\right)\psi - \epsilon\right]}\left[\left(1+\frac{1}{\gamma}\right)\psi - \epsilon\right]\sigma_a = \sigma_b \tag{31}$$

Finally, as the validator c has no neighbors, it also satisfies the condition, whence the graph indeed satisfies (17). We now consider an initial shock $D := \{a\}$ to the graph. Because

$$\frac{\sigma_a}{\sigma_V} = \frac{\sigma_a}{\sigma_a/\psi} = \psi, \tag{32}$$

the shock $D \in \mathbb{D}_\psi$ constitutes a ψ fraction of the total stake as desired. The attack $(\{x\}, \{b\})$ is a valid attack on $G \searrow D$. To see this, notice first that $(\{x\}, \{b\})$ is an attacking coalition on $G \searrow D$ as $\{b\} = N_{G\searrow D}\{x\}$. Furthermore, $\{b\}$ is incentivized to attack since

$$\pi_x - \sigma_b = \frac{\left(1+\frac{1}{\gamma}\right)\psi - \epsilon}{1+\gamma} \cdot \sigma_V - \sigma_b \tag{33}$$

$$= \left[\frac{1+\frac{1}{\gamma}-\frac{\epsilon}{\psi}}{1+\gamma} - \left(\frac{1}{\gamma} - \frac{\epsilon}{\psi}\right)\right]\sigma_a \tag{34}$$

$$= \left[\frac{\left(1+\frac{1}{\gamma}\right)\psi + \gamma\epsilon}{(1+\gamma)\psi} - \frac{1}{\gamma}\right]\sigma_a \tag{35}$$

$$= \frac{\gamma\epsilon\sigma_a}{(1+\gamma)\psi} > 0 \tag{36}$$

whence Equation (2) is satisfied for the pair $(\{x\}, \{b\})$. It follows that

$$R_\psi(G) \geq \frac{\sigma_a + \sigma_b}{\sigma_V} = \frac{\left(1+\frac{1}{\gamma}-\frac{\epsilon}{\psi}\right)\sigma_a}{\sigma_a/\psi} = \left(1+\frac{1}{\gamma}\right)\psi - \epsilon \tag{37}$$

as desired. ◀

5 Local Security

The bound in Theorem 4 on the worst-possible stake loss from cascading attacks is reassuring from a global perspective, but less so from the perspective of one or a small number of services who would like an assurance that they will not be among those affected by such attacks. Beginning with this section, we focus on a specific coalition of services $C \subseteq S$ that seeks to insulate their shared security $\Gamma(C)$ against shocks and resulting cascading attacks that may come about due to the decisions of other services and validators. Formally, we denote by

$$\Gamma(C) := \{v \in V \mid N_G\{v\} \subseteq C\} \tag{38}$$

the set of validators that exclusively provide security for services in C.

Worst-case stake loss (local version)

As before, we first suppose that an initial shock affects the restaking graph. Whereas previously, we considered shocks for which the total stake in the shock was bounded, we now consider shocks for which the total stake that impacts the exclusive security of some coalition of services C (i.e., stake that secures services from C and only services from C) is bounded. Formally, for any coalition of services $C \subseteq S$ within some restaking graph G, we let

$$\mathbb{D}_\psi(C,G) := \left\{ D \subseteq V \mid \frac{\sigma_{D\cap\Gamma(C)}}{\sigma_{\Gamma(C)}} \leq \psi \right\} \tag{39}$$

denote the set of all validator coalitions that provide at most ψ stake to the aggregate security of the coalition of services C. Notice that shocks $D \in \mathbb{D}_\psi(C,G)$ may have much more total stake σ_D than a ψ fraction of the graph. We are instead only guaranteed that the impact of the shock on stake that is being used exclusively for members in C is small. We are now interested in the potential cascading losses that can affect the stake that is exclusively utilized by the coalition C after a shock occurs that destroys at most ψ stake from the aggregate security of C. Formally, we study the quantity

$$R_\psi(C,G) := \underbrace{\psi}_{\text{Initial Shock}} + \max_{D\in\mathbb{D}_\psi(C,G)} \max_{(A_1,B_1),\ldots,(A_T,B_T)\in\mathcal{C}(G\searrow D)} \underbrace{\frac{\sigma_{\bigcup_{t=1}^T B_t\cap\Gamma(C)}}{\sigma_{\Gamma(C)}}}_{\text{Stake lost from cascading attacks}} \tag{40}$$

Local security conditions

We seek sufficient conditions on a restaking graph G that guarantee a nontrivial upper bound on $R_\psi(C,G)$. Ideally, the sufficient condition would depend only the neighborhood/choices of the coalition C to provide a guarantee that holds regardless of the choices made by other validators and services (i.e., the services of C can attempt to "control their own destiny" by ensuring that the locally defined sufficient condition holds). Formally, given a restaking graph $G = (S,V,E,\pi,\sigma,\alpha)$ and a coalition of services $C \subseteq S$, we call a restaking graph $G' = (S',V',E',\pi',\sigma',\alpha')$ a *C-local variant* of G if C cannot distinguish G' from G on the basis of local information: $C \subseteq S'$, $N_G C = N_{G'} C$, and

$$(\pi_s,\alpha_s) = (\pi'_s,\alpha'_s) \qquad \forall s \in C \tag{41}$$

$$(\sigma_v, N_G\{v\}) = (\sigma'_v, N_{G'}\{v\}) \qquad \forall v \in N_G C \tag{42}$$

We then define a *local security condition* $f : (C,G) \mapsto \{0,1\}$ to be a Boolean function that takes as input a restaking graph G and a coalition of services $C \subseteq S$ such that $f(C,G)$ must be equal to $f(C,G')$ for all C-local variants G'. The intuition behind this definition is that the condition should only depend on service-level information (e.g. profits from corruption, security thresholds) for services in the coalition, and validator-level information for validators in $N_G C$.

Local security impossibility

Unfortunately, without further restrictions on the attacks under consideration (like those defined later in this section), it is impossible to construct any nontrivial local security condition that yields any nontrivial upper bound on $R_\psi(C,G)$.

▶ **Theorem 8.** *For any local security condition f, any secure restaking graph G and coalition of services $C \subseteq S$ such that $f(C,G) = 1$, there exists a secure C-local variant G' of G such that $R_0(C,G') = 1$.*

Proof. Take any f, C, and secure G such that $f(C,G) = 1$. Define

$$\Delta := \sigma_{N_G C} - \pi_C \tag{43}$$

to be the total overcollateralization of C in aggregate. Next, we define G' to be an augmented version of G, where we add a new service s^* that has a profit from corruption $\pi_{s^*} = \Delta + 2\epsilon$ where $\epsilon > 0$. We further add 2 validators a and b to the graph who are adjacent only to s^*. We let $\sigma_a := \Delta + \epsilon$ and $\sigma_b := \epsilon$. As $\sigma_a + \sigma_b \geq \pi_{s^*}$, the graph G' must be secure as G was secure. Next, notice that as the validator a is not path-connected to C, G' must be a C-local variant of G. However, by construction, the attack $(C \cup \{s^*\}, N_G C \cup \{b\})$ is valid on the graph $G' \searrow \{a\}$. It follows that $R_0(C, G') = 1$. ◀

Stable attacks

While the above impossibility appears to be quite strong, it is somewhat contrived. At the heart of the impossibility is that under the definition of a valid attack (i.e. Equations (1) and (2)), not every validator must be productive in carrying out the attack. There may be a subset of validators in the attack that can yield more net profit than the full coalition. In what follows, we show that if we assume that malicious validator coalitions will choose to add others to their ranks only if it is profitable for them in net to do so, then a local security condition with guarantees similar to those in Theorem 4 does indeed exist. Formally, for $A \subseteq S$ and $B \subseteq V$, we say that an attack (A, B) is *stable* if it is valid (i.e. Equations (1) and (2) hold), and for all $A' \subseteq A$ and $B' \subseteq B$ such that (A', B') is valid,

$$\sigma_{B \setminus B'} < \pi_{A \setminus A'} \tag{44}$$

Intuitively, if (44) did not hold, then the validators of B' would be better off ditching those in $B \setminus B'$ and attacking only the services in A'.

We say that a disjoint sequence $(A_1, B_1), \ldots, (A_T, B_T) \in 2^S \times 2^V$ is a cascade of stable attacks on a restaking graph $G = (S, V, E, \pi, \sigma, \alpha)$ if for each $t \in [T]$, (A_t, B_t) is a stable attack on $G \searrow \bigcup_{i=1}^{t-1} B_i$. We denote by $\mathcal{S}(G)$ the set of all such sequences of cascading stable attacks. In light of Theorem 8, we redefine our notion of worst-case stake loss using stable attacks:

$$R_\psi(C, G) := \psi + \max_{D \in \mathbb{D}_\psi(C,G)} \max_{(A_1,B_1),\ldots,(A_T,B_T) \in \mathcal{S}(G \searrow D)} \frac{\sigma_{\bigcup_t B_t \cap \Gamma(C)}}{\sigma_{\Gamma(C)}} \tag{45}$$

Unlike valid attacks, unions of sequences of stable cascading attacks need not be stable (which will complicate the proof of Theorem 10 in the next section).

▷ Claim 9. There exists a restaking graph G and a sequence $(A_1, B_1), (A_2, B_2) \in \mathcal{S}(G)$ such that $(A_1 \cup A_2, B_1 \cup B_2) \notin \mathcal{S}(G)$.

Proof. Consider restaking graph where there are two services $S = \{x, y\}$ and two validators $V = \{a, b\}$ where both validators are restaking in both services. Furthermore, $\pi_x = \pi_y = 2$, $\alpha_x = \alpha_y = \frac{1}{2}$, and $\sigma_a = \sigma_b = 1$. In this case, the sequence $(\{x\}, \{a\}), (\{y\}, \{b\}) \in \mathcal{S}(G)$. However, $(\{x, y\}, \{a, b\}) \notin \mathcal{S}(G)$ because the attack $(\{x, y\}, \{a\})$ is a valid attack. ◁

6 A Local Security Condition for Stable Attacks

In this section, we give a family of local security conditions that yield guarantees on the local worst-case stake loss $R_\psi(C, G)$ that resemble our result from Theorem 4 for global security. In Theorems 4 and 7, we showed that security with γ-slack was necessary and sufficient in order to obtain a $\left(1 + \frac{1}{\gamma}\right)\psi$ upper-bound on $R_\psi(G)$. In other words, it was both necessary and sufficient to ensure that all attacking coalitions (A, B) were overcollateralized

multiplicatively by a factor of $(1+\gamma)$. Our condition for local security is similar: we must ensure that certain *attack headers* (X, Y) are overcollateralized multiplicatively by a factor of $(1+\gamma)$.

Attack header

Formally, we say that (X, Y) is an *attack header*, for $X \subseteq S$ and $Y \subseteq \Gamma(X)$, if there exists a set of validators $B \subseteq V$ satisfying

$$B \cap \Gamma(X) = \emptyset \tag{46}$$

such that $(X, B \cup Y)$ is an attacking coalition. In other words, (X, Y) is an attack header if Y can be appended to a collection of validators B that may be slashed without attacking services in X to form an attacking coalition that attacks the services X. An attacking coalition is a special case of an attack header (in which B can be taken as $\emptyset$).

▶ **Theorem 10.** *Let $G = (S, V, E, \pi, \sigma, \alpha)$ be a restaking graph and $C \subseteq S$ be a coalition of services. If, for all attack headers (X, Y) where $X \subseteq C$,*

$$(1+\gamma)\pi_X \leq \sigma_Y \tag{47}$$

then $R_\psi(C, G) < (1 + \frac{1}{\gamma})\psi$. Furthermore, the Boolean function that checks whether Equation (47) holds for all attack headers is a local security condition.

Proof. Let $D \in \mathbb{D}_\psi(C, G)$ and $(A_1, B_1), \ldots, (A_T, B_T) \in \mathcal{S}(G \searrow D)$ be arbitrary. For each $t \in [T]$, define

$$L_t := B_t \cap \Gamma(C) \tag{48}$$

to be the set of all validators exclusively securing C that were lost in the t^{th} attack. Next, define

$$A'_t := \left\{ s \in A_t \mid \sigma_{B_t \setminus L_t} \geq \alpha_s \cdot \sigma_{N_G\{s\} \setminus \left(\bigcup_{i=1}^{t-1} B_i \cup D\right)} \right\} \tag{49}$$

to be the maximal set of services such that $(A'_t, B_t \setminus L_t)$ is an attacking coalition on $G \searrow \left(\bigcup_{i=1}^{t-1} B_i \cup D\right)$. Notice also that because (A_t, B_t) is a stable attack on $G \searrow \left(\bigcup_{i=1}^{t-1} B_i \cup D\right)$, we must have that $A_t \setminus A'_t$ is nonempty, and furthermore must be a subset of C.

▷ Claim 11.

$$\sigma_{\bigcup_t B_t \cap \Gamma(C)} < \pi_{\bigcup_t A_t \setminus A'_t} \tag{50}$$

Proof. From stability, we have that

$$\sigma_{L_t} = \sigma_{B_t \setminus (B_t \setminus L_t)} < \pi_{A_t \setminus A'_t} \tag{51}$$

To see why this holds, notice that there are two cases. If the attacking coalition $(A'_t, B_t \setminus L_t)$ is a valid attack, then the inequality follows directly from the stability definition. If instead $(A'_t, B_t \setminus L_t)$ is not a valid attack despite being an attacking coalition, the inequality must still hold because the original attack (A_t, B_t) is valid and therefore satisfies Equation (2). Iterating over t, we find that

$$\sigma_{\bigcup_t B_t \cap \Gamma(C)} = \sum_{t=1}^{T} \sigma_{L_t} < \sum_{t=1}^{T} \pi_{A_t \setminus A'_t} = \pi_{\bigcup_t A_t \setminus A'_t} \tag{52}$$

◁

▷ Claim 12. $\left(\bigcup_t (A_t \setminus A'_t), (\bigcup_t B_t \cup D) \cap \Gamma(C)\right)$ is an attack header on G, and $\bigcup_t (A_t \setminus A'_t) \subseteq C$.

Proof. Because each $A_t \setminus A'_t \subseteq C$, we must also have that $\bigcup_t A_t \setminus A'_t \subseteq C$ as well. Next, by applying Corollary 3 noting the disjointness of the (A_t, B_t), we have that

$$\left(\bigcup_t A_t, \bigcup_t B_t\right) = \left(\bigcup_t (A_t \setminus A'_t) \cup \bigcup_t A'_t, \quad \bigcup_t B_t\right) \tag{53}$$

must be an attacking coalition on $G \searrow D$. In particular, we must have that $\left(\bigcup_t (A_t \setminus A'_t), \bigcup_t B_t\right)$ is an attacking coalition on $G \searrow D$, whence by Lemma 3 we have that $\left(\bigcup_t (A_t \setminus A'_t), \bigcup_t B_t \cup D\right)$ is an attacking coalition on G. Rewriting the above as

$$\left(\bigcup_t (A_t \setminus A'_t), \quad \left[\left(\bigcup_t B_t \cup D\right) \cap \Gamma(C)\right] \cup \left[\left(\bigcup_t B_t \cup D\right) \setminus \Gamma(C)\right]\right) \tag{54}$$

and noting by construction that

$$\left[\left(\bigcup_t B_t \cup D\right) \setminus \Gamma(C)\right] \cap \Gamma(C) = \emptyset, \tag{55}$$

we find that $\left(\bigcup_t (A_t \setminus A'_t), (\bigcup_t B_t \cup D) \cap \Gamma(C)\right)$ is an attack header on G as desired. ◁

Putting these claims together yields the desired result. By Claim 12 and Equation (47), we find that

$$(1+\gamma)\pi_{\bigcup_t A_t \setminus A'_t} \leq \sigma_{(\bigcup_t B_t \cup D) \cap \Gamma(C)} \leq \sigma_{\bigcup_t B_t \cap \Gamma(C)} + \psi \cdot \sigma_{\Gamma(C)} \tag{56}$$

Adding in Claim 11, we find that

$$(1+\gamma)\sigma_{\bigcup_t B_t \cap \Gamma(C)} < \sigma_{\bigcup_t B_t \cap \Gamma(C)} + \psi \cdot \sigma_{\Gamma(C)} \tag{57}$$

$$\implies \frac{\sigma_{\bigcup_t B_t \cap \Gamma(C)}}{\sigma_{\Gamma(C)}} < \frac{\psi}{\gamma} \tag{58}$$

$$\implies R_\psi(C, G) < \left(1 + \frac{1}{\gamma}\right)\psi \tag{59}$$

as desired. To see that the Boolean function that checks whether Equation (47) holds for all attack headers is a local security condition, observe that it suffices to check that for all $X \subseteq C$ and $Y \subseteq N_G C \cap \Gamma(C) = \Gamma(C)$ such that $(X, Y \cup N_G C \setminus \Gamma(C))$ is an attacking coalition, Equation (47) holds. Thus, for any restaking graph G and C-local variant G', this function will evaluate to the same output. ◀

The condition in (47) can be checked via enumeration, although the time required to do so grows exponentially in the number of services in C and the number of validators that contribute security exclusively to services in C. In the event that this is a prohibitive amount of computation, the same guarantee holds under a stronger, easily checked local analog of the EigenLayer sufficient condition (3) which, in effect, treats as malicious all validators that contribute security to any services outside of C.

▶ **Corollary 13.** *For any restaking graph $G = (S, V, E, \pi, \sigma, \alpha)$ and coalition of services $C \subseteq S$, satisfaction of the local condition*

$$\sum_{s \in N_G\{v\}} \frac{\sigma_v}{\sigma_{N_G(\{s\})}} \cdot \frac{(1+\gamma)\pi_s}{\alpha'_s} \leq \sigma_v \tag{60}$$

for all $v \in N_G C$ with $N_G\{v\} \subseteq C$ guarantees that $R_\psi(C, G) < \left(1 + \frac{1}{\gamma}\right)\psi$, where for each $s \in C$,

$$\alpha'_s := \alpha_s - \frac{\sigma_{N_G\{s\} \setminus \Gamma(C)}}{\sigma_{N_G\{s\}}} \tag{61}$$

Proof. This follows from Theorem 10 and Claim 1. Notice that to guarantee that the condition (47) from Theorem 10 holds for all attack headers (X, Y), it suffices to guarantee that no valid attacks exist, when (i) all profits from corruption for services in C are inflated by a multiplicative factor of $(1 + \gamma)$, and (ii) the fraction of stake required to corrupt a given service is offset by the fraction of stake in that service that is also restaking for services that do not belong to C. As Claim 1 provides a sufficient condition to guarantee the existence of no valid attacks, Equation (60) will guarantee that all attack headers are overcollateralized by a multiplicative factor of $(1 + \gamma)$. ◀

7 Lower Bounds for Local Security

We next show senses in which Theorem 10 (and more strongly, Corollary 13) is tight.

▶ **Corollary 14.** *For any $\psi, \gamma, \epsilon > 0$ such that*

$$0 \leq \left(1 + \frac{1}{\gamma}\right)\psi - \epsilon \leq 1, \tag{62}$$

there exists a restaking graph G that satisfies the condition (60) from Corollary 13 but has $R_\psi(C, G) \geq \left(1 + \frac{1}{\gamma}\right)\psi - \epsilon$.

Proof. Notice that if we take $C = S$, the condition (60) from Corollary 13 is identical to the condition (17) from Corollary 5. Furthermore, $R_\psi(S, G)$ is the same as $R_\psi(G)$ except for the fact that $R_\psi(S, G)$ considers only stable attacks. Repeating the argument from Theorem 7, and noting that the attack given in the proof of that result is stable, we obtain the desired result. ◀

▶ **Theorem 15.** *For any $\gamma > 0$, there exists a graph restaking graph $G = (S, V, E, \pi, \sigma, \alpha)$ that satisfies (17) from Corollary 5, but there exists a $C \subseteq S$ such that $R_0(C, G) = 1$.*

Proof. Let there be three services $S = \{x, y, z\}$, three validators $V = \{a, b, c\}$, and edges as follows:

$$E := \{(x, a), (x, b), (y, b), (y, c), (z, c), (z, a)\}. \tag{63}$$

Next, let $\alpha_s = 1$ for all $s \in S$, and let $\pi_x = \pi_y = \pi_z =: \pi > 0$ be an arbitrary positive constant. Next pick σ_a such that

$$\sigma_a < 2\pi, \tag{64}$$

let $\sigma_b := 2(1+\gamma)\pi$, and let $\sigma_c := 2(1+\gamma)\pi$. This graph satisfies (17) from Corollary 5 as

$$\frac{\sigma_a}{\sigma_a+\sigma_b}\cdot(1+\gamma)\pi + \frac{\sigma_a}{\sigma_a+\sigma_c}\cdot(1+\gamma)\pi < (1+\gamma)\pi\left(\frac{1}{\sigma_b}+\frac{1}{\sigma_c}\right)\sigma_a = \sigma_a \tag{65}$$

$$\frac{\sigma_b}{\sigma_a+\sigma_b}\cdot(1+\gamma)\pi + \frac{\sigma_b}{\sigma_b+\sigma_c}\cdot(1+\gamma)\pi < \frac{3}{2}(1+\gamma)\pi < \sigma_b \tag{66}$$

$$\frac{\sigma_c}{\sigma_a+\sigma_c}\cdot(1+\gamma)\pi + \frac{\sigma_c}{\sigma_b+\sigma_c}\cdot(1+\gamma)\pi < \frac{3}{2}(1+\gamma)\pi < \sigma_c \tag{67}$$

Next, let $C := \{x, z\}$, and observe that the shock $D = \{b, c\}$ is an element of $\mathbb{D}_0(C, G)$. However, because $\sigma_a < \pi_x + \pi_z = 2\pi$, we must have that $(\{x, z\}, \{a\})$ is a stable attack on $G \searrow D$. As $\{a\} = \Gamma(C)$, it follows that $R_0(C, G) = 1$. ◀

8 Long Cascades

Cascade Structure

Although Corollary 3 shows that all long cascades can be made into a short, one-step attack, it is arguably more dangerous if large attacks can be made through long cascades of small attacks that each require less coordination. In this section, we show how adding γ-slack also enables us to upper-bound the length of a cascade in the worst case. Our results depend on what we call the *reference depth* of a cascading sequence of attacks. Formally, for a given restaking graph G and coalition of validators B_0, we say that a valid cascade of attacks $(A_1, B_1), \ldots, (A_T, B_T)$ on $\mathcal{C}(G \searrow B_0)$ has reference depth k if

$$k = \max\{i \in [T] \mid \exists t \in [T] \text{ s.t. } N_G(A_t) \cap B_{t-i} \neq \emptyset\} \tag{68}$$

In other words, a cascading sequence has reference depth k if the services attacked in a given time step are not affected by validators that were slashed more than k steps previous.

▶ **Theorem 16.** *Suppose that a restaking graph $G = (S, V, E, \pi, \sigma, \alpha)$ is secure with γ-slack for some $\gamma > 0$. Let $\epsilon = \min_{v \in V} \sigma_v$ denote the minimum stake held by a validator. Then, for any $\psi > 0$, $B_0 \in \mathbb{D}_\psi(G)$, and $(A_1, B_1), \ldots, (A_T, B_T) \in \mathcal{C}(G \searrow B_0)$ with reference depth k,*

$$T < k\left(1 + \log_{1+\gamma}\frac{\psi \cdot \sigma_V}{\epsilon\gamma}\right) \tag{69}$$

Proof. Because $(A_1, B_1), \ldots, (A_T, B_T) \in \mathcal{C}(G \searrow B_0)$ has reference depth k, it follows that if we define

$$A'_i := \bigcup_{t=k(i-1)+1}^{ki} A_t \tag{70}$$

$$B'_i := \bigcup_{t=k(i-1)+1}^{ki} B_t \tag{71}$$

$$B'_0 := B_0 \tag{72}$$

for $i \in \{1, \ldots, \lceil T/k \rceil\}$, we must have that for all $j < i - 1$,

$$N_G A'_i \cap B'_j = \emptyset \tag{73}$$

Furthermore, by Corollary 3, we must have that the sequence $(A'_i, B'_i) \in \mathcal{C}(G \searrow B_0)$. Applying Corollary 3 again, we further have that for every i, $\left(\bigcup_{j=i+1}^{\lceil T/k \rceil} A'_i, \bigcup_{j=i+1}^{\lceil T/k \rceil} B'_i\right)$ is a valid attack on $G \searrow \bigcup_{j=0}^{i} B'_i$. It follows by Equation (2) that for any $i \in \{0, \ldots, \lceil T/k \rceil - 1\}$,

$$\pi_{\bigcup_{j=i+1}^{\lceil T/k \rceil} A'_j} > \sigma_{\bigcup_{j=i+1}^{\lceil T/k \rceil} B'_j} = \sum_{j=i+1}^{\lceil T/k \rceil} \sigma_{B'_j} \tag{74}$$

Next, noting that

$$N_G \bigcup_{j=i+1}^{\lceil T/k \rceil} A'_i \cap \bigcup_{j=1}^{i-1} B'_i = \emptyset \tag{75}$$

by Equation (73), it follows that indeed $\left(\bigcup_{j=i+1}^{\lceil T/k \rceil} A'_i, \bigcup_{j=i+1}^{\lceil T/k \rceil} B'_i\right)$ is also a valid attack on $G \searrow B'_i$. Applying Lemma 2, we then find that $\left(\bigcup_{j=i+1}^{\lceil T/k \rceil} A'_i, B'_i \cup \bigcup_{j=i+1}^{\lceil T/k \rceil} B'_i\right)$ is an attacking coalition on the original graph G. Because G is secure with γ-slack, Equation (11) must now hold on this pair, whence for all $i \in \{0, \ldots, \lceil T/k \rceil - 1\}$,

$$(1+\gamma)\pi_{\bigcup_{j=i+1}^{\lceil T/k \rceil} A'_j} \leq \sigma_{\bigcup_{j=i}^{\lceil T/k \rceil} B'_j} = \sigma_{B'_i} + \sum_{j=i+1}^{\lceil T/k \rceil} \sigma_{B'_j} \tag{76}$$

Putting this together with Equation (74), we have that for all $i \in \{0, \ldots, \lceil T/k \rceil - 1\}$,

$$(1+\gamma) \sum_{j=i+1}^{\lceil T/k \rceil} \sigma_{B'_j} < \sigma_{B'_i} + \sum_{j=i+1}^{\lceil T/k \rceil} \sigma_{B'_j} \tag{77}$$

$$\implies \sigma_{B'_i} > \gamma \sum_{j=i+1}^{\lceil T/k \rceil} \sigma_{B'_j} \tag{78}$$

It can be shown inductively from Equation (78) that the sequence $\sigma_{B'_i} > X_i$, where we define the sequence X_i by

$$X_{\lceil T/k \rceil} := \epsilon \tag{79}$$

$$X_i := \gamma \sum_{j=i+1}^{\lceil T/k \rceil} X_j \qquad\qquad i \in \{0, \ldots, \lceil T/k \rceil - 1\} \tag{80}$$

Solving, we find that

$$X_0 = \epsilon\gamma(1+\gamma)^{\lceil T/k \rceil - 1} \tag{81}$$

It then follows as desired that

$$\psi \cdot \sigma_V = \sigma_{B_0} > \epsilon\gamma(1+\gamma)^{T/k-1} \implies T < k\left(1 + \log_{1+\gamma} \frac{\psi \cdot \sigma_V}{\epsilon\gamma}\right) \tag{82}$$

◀

References

1 Daron Acemoglu, Asuman Ozdaglar, and Alireza Tahbaz-Salehi. Networks, shocks, and systemic risk. Technical report, National Bureau of Economic Research, 2015.

2 Daron Acemoglu, Asuman Ozdaglar, and Alireza Tahbaz-Salehi. Systemic risk and stability in financial networks. *American Economic Review*, 105(2):564–608, 2015.

3 Carol Alexander. Leveraged restaking of leveraged staking: What are the risks? *Available at SSRN 4840805*, 2024.

4 Stefano Battiston, Guido Caldarelli, Robert M May, Tarik Roukny, and Joseph E Stiglitz. The price of complexity in financial networks. *Proceedings of the National Academy of Sciences*, 113(36):10031–10036, 2016.

5 Markus K Brunnermeier, Gary Gorton, and Arvind Krishnamurthy. Risk topography. *Nber macroeconomics annual*, 26(1):149–176, 2012.

6 Chen Chen, Garud Iyengar, and Ciamac C Moallemi. An axiomatic approach to systemic risk. *Management Science*, 59(6):1373–1388, 2013. doi:10.1287/MNSC.1120.1631.

7 Soubhik Deb, Robert Raynor, and Sreeram Kannan. Stakesure: Proof of stake mechanisms with strong cryptoeconomic safety. *arXiv preprint*, 2024. doi:10.48550/arXiv.2401.05797.

8 Douglas W Diamond and Philip H Dybvig. Bank runs, deposit insurance, and liquidity. *Journal of political economy*, 91(3):401–419, 1983.

9 Larry Eisenberg and Thomas H Noe. Systemic risk in financial systems. *Management Science*, 47(2):236–249, 2001. doi:10.1287/MNSC.47.2.236.9835.

10 Zachary Feinstein, Birgit Rudloff, and Stefan Weber. Measures of systemic risk. *SIAM Journal on Financial Mathematics*, 8(1):672–708, 2017. doi:10.1137/16M1066087.

11 Paul Glasserman and H Peyton Young. Contagion in financial networks. *Journal of Economic Literature*, 54(3):779–831, 2016.

12 Subhash Khot and Rishi Saket. Hardness of bipartite expansion. In *24th Annual European Symposium on Algorithms (ESA 2016)*. Schloss-Dagstuhl – Leibniz Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.ESA.2016.55.

13 Eduard Kromer, Ludger Overbeck, and Katrin Zilch. Systemic risk measures on general measurable spaces. *Mathematical Methods of Operations Research*, 84:323–357, 2016. doi:10.1007/S00186-016-0545-1.

14 Tarun Chitra Mike Neuder. The risks of lrts, 2024. URL: https://ethresear.ch/t/the-risks-of-lrts/18799.

15 EigenLayer Team. Eigenlayer: The restaking collective, 2023. URL: https://docs.eigenlayer.xyz/overview/whitepaper.

Confusion Matrix Design for Downstream Decision-Making

Yiding Feng
Department of Industrial Engineering and Decision Analytics,
Hong Kong University of Science and Technology, Hong Kong

Wei Tang
Department of Decisions, Operations and Technology,
Chinese University of Hong Kong, Hong Kong

Abstract

We initiate the study of *confusion matrix design*. In this problem, an algorithm designer needs to generate a machine learning model (for a classification task from contexts to labels) which makes predictions for a population of downstream decision makers. The prediction accuracy of the machine learning model is characterized by its confusion matrix, which is a stochastic matrix where each entry encodes the probability of predicting the true label to another label. Each downstream decision maker faces a separate optimization task and will decide his binary action based on his own context, realized prediction given his context, and the confusion matrix selected by the algorithm designer. Decision makers are heterogeneous, as they may hold different contexts. Both the decision makers and the algorithm designer will obtain utilities that are determined by the actions the decision makers take, and their true labels. The goal of the algorithm designer is to design a public confusion matrix that is used for all decision makers subject to some feasibility constraints in order to maximize her net utility. We consider a general class of net utility functions, which could be a combination of both decision makers' utilities and the algorithm designer's utility. Classic outcome-independent utility and utilitarian/Nash/egalitarian social welfare are all special cases of our net utility formulation.

We study the above problem through an information design framework, where we view training machine learning model as designing an information structure (signaling scheme) subject to some specific constraints motivated by the machine learning literature. By building the connection to the *public persuasion with heterogeneous priors*, we design convex programming-based algorithms that compute the optimal confusion matrix subject to (i) post-processing constraints and (ii) receiver operating characteristic (ROC) constraints in polynomial time, respectively. Besides the computational results, we also obtain analytical structural results and numerical results for the special cases of outcome-independent utility and social-aware utility, by utilizing the convex programming-based characterization of the optimal confusion matrix.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory; Theory of computation → Machine learning theory

Keywords and phrases Confusion Matrix, Downstream Decision-making, Public Persuasion, Human-AI Interaction, Information Design

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.49

Category Extended Abstract

Related Version *Full Version*: `https://doi.org/10.2139/ssrn.4950758` [1]

Acknowledgements We thank the ITCS reviewers for the helpful and constructive comments.

References

1 Yiding Feng and Wei Tang. Confusion matrix design for downstream decision-making. *SSRN*, 2024. Full version of this abstract.



© Yiding Feng and Wei Tang;
licensed under Creative Commons License CC-BY 4.0



Fully Characterizing Lossy Catalytic Computation

Marten Folkertsma
CWI, Amsterdam, The Netherlands
QuSoft, Amsterdam, The Netherlands

Ian Mertz
University of Warwick, UK

Florian Speelman
University of Amsterdam, The Netherlands

Quinten Tupker
CWI, Amsterdam, The Netherlands

Abstract

A *catalytic machine* is a model of computation where a traditional space-bounded machine is augmented with an additional, significantly larger, "catalytic" tape, which, while being available as a work tape, has the caveat of being initialized with an arbitrary string, which must be preserved at the end of the computation. Despite this restriction, catalytic machines have been shown to have surprising additional power; a logspace machine with a polynomial length catalytic tape, known as *catalytic logspace* (CL), can compute problems which are believed to be impossible for L.

A fundamental question of the model is whether the catalytic condition, of leaving the catalytic tape in its exact original configuration, is robust to minor deviations. This study was initialized by Gupta et al. (2024), who defined *lossy catalytic logspace* ($\mathsf{LCL}[e]$) as a variant of CL where we allow up to e errors when resetting the catalytic tape. They showed that $\mathsf{LCL}[e] = \mathsf{CL}$ for any $e = O(1)$, which remains the frontier of our understanding.

In this work we completely characterize lossy catalytic space ($\mathsf{LCSPACE}[s, c, e]$) in terms of ordinary catalytic space ($\mathsf{CSPACE}[s, c]$). We show that

$$\mathsf{LCSPACE}[s, c, e] = \mathsf{CSPACE}[\Theta(s + e \log c), \Theta(c)]$$

In other words, allowing e errors on a catalytic tape of length c is equivalent, up to a constant stretch, to an equivalent errorless catalytic machine with an additional $e \log c$ bits of ordinary working memory.

As a consequence, we show that for any e, $\mathsf{LCL}[e] = \mathsf{CL}$ implies $\mathsf{SPACE}[e \log n] \subseteq \mathsf{ZPP}$, thus giving a barrier to any improvement beyond $\mathsf{LCL}[O(1)] = \mathsf{CL}$. We also show equivalent results for *non-deterministic* and *randomized catalytic space*.

2012 ACM Subject Classification Theory of computation → Complexity theory and logic; Theory of computation → Complexity classes

Keywords and phrases Space complexity, catalytic computation, error-correcting codes

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.50

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2024/138/`

Funding *Marten Folkertsma*: Supported by the Dutch Ministry of Economic Affairs and Climate Policy (EZK), as part of the Quantum Delta NL program.
Ian Mertz: Supported by Royal Society University Research Fellowship URF\R1\191059.
Florian Speelman: Supported by the Dutch Ministry of Economic Affairs and Climate Policy (EZK), as part of the Quantum Delta NL program, and the project Divide and Quantum "D&Q" NWA.1389.20.241 of the program "NWA-ORC", which is partly funded by the Dutch Research Council (NWO).
Quinten Tupker: Supported by the Dutch National Growth Fund (NGF), as part of the Quantum Delta NL program.

Acknowledgements We acknowledge useful conversations with Swastik Kopparty and Geoffrey Mon about error correction codes and how to apply them.



© Marten Folkertsma, Ian Mertz, Florian Speelman, and Quinten Tupker;
licensed under Creative Commons License CC-BY 4.0



1 Introduction

1.1 Catalytic computation

Within space-bounded computation, the *catalytic computing* framework, first introduced by Buhrman, Cleve, Koucký, Loff, and Speelman [4], models the question of whether or not full memory can be a computational resource. Their main object of study is a *catalytic logspace* (CL) machine, in which a traditional logspace-bounded Turing machine is given access to a second work tape, polynomial in length, called the catalytic tape; while this tape is exponentially longer than the logspace work tape, it is already full with some string τ at the outset, and this string τ must be preserved by the overall computation.

Surprisingly, [4] show that CL can be much more powerful than L, with the catalytic tape being at least as powerful a resource as non-determinism ($\mathsf{NL} \subseteq \mathsf{CL}$), randomness ($\mathsf{BPL} \subseteq \mathsf{CL}$), and more ($\mathsf{TC}^1 \subseteq \mathsf{CL}$). They also showed that its power is nevertheless limited and falls far short PSPACE, namely $\mathsf{CL} \subseteq \mathsf{ZPP}$.

This work spawned a long sequence of explorations of the power of catalytic space. Given the base model of CL there are many possible variations and structural questions to be asked, such as the power of randomness [11, 6], non-determinism [5], non-uniformity [21, 23, 9, 10], and other variants [15, 2]. There have also been many works connecting the catalytic framework to broader questions in complexity theory, such as space-bounded derandomization [22, 14, 19], as well as adaptations of catalytic techniques to solve longstanding open questions such as compositional upper bounds for space [7, 8, 10] (see [18, 20] for surveys on the topic).

1.2 Lossy catalytic computation

Besides these more standard structural questions, there are also catalytic variants which are more specific to the catalytic space restriction. In particular, Gupta et al. [16] initiated the study of *lossy* catalytic computing, wherein the catalytic tape need not be exactly reset to its initial configuration. This model, which we refer to as LCSPACE, essentially asks how robust the core definition of catalytic space is to seemingly small relaxations; for example, in the *quantum* setting some computation error (albeit of a different form) is necessary for converting between different definitions based on allowed operations.

To begin, note that CL with $e \leq \text{poly}(n)$ errors trivially contains the class $\mathsf{SPACE}[e]$ by simply erasing the first e bits of the catalytic tape and using them as free memory. Because we have not managed to prove that any space-bounded class beyond L which is contained in ZPP, we should not expect to be able to prove CL is the same as CL with $e = \omega(\log n)$ errors. The question, then, is to understand where, in the range of $e = 0$ to $e = O(\log n)$, is the acceptable number of errors that CL can provably tolerate.

As an initial answer to the previous question, [16] show that CL gains no additional power from allowing any constant number of errors on the catalytic tape, i.e., $\mathsf{LCL}[O(1)] = \mathsf{CL}$. This remains the frontier of our knowledge, and Mertz [20] posed it as an open question to improve this result to any superconstant number of errors, or, alternatively, to provide evidence against being able to prove such a collapse.[1] Recently, Cook et al. [6] showed that a different error-prone model, namely *randomized* CL, is no more powerful than the base CL model.

[1] We cannot expect an unconditional separation between CL and any LCL, as even separating PSPACE from e.g. $\mathsf{TC}^1 (\subseteq \mathsf{CL})$ remains wide open.

1.3 Our results

In this work we completely characterize lossy catalytic space in terms of ordinary catalytic space. Let $\mathsf{CSPACE}[s, c]$ denote catalytic machines with free space s and catalytic space c, and let $\mathsf{LCSPACE}[s, c, e]$ be the same with up to e errors allowed in resetting the catalytic tape. We show that these e errors are equivalent to an additional $e \log c$ free bits of memory, up to constant factor losses.

▶ **Theorem 1.** *Let $s := s(n), c := c(n), e := e(n)$ be such that $e \leq c^{1-\Omega(1)}$. Then*

$$\mathsf{LCSPACE}[\Theta(s), \Theta(c), e] = \mathsf{CSPACE}[\Theta(s + e \log c), \Theta(c)]$$

Besides characterizing $\mathsf{LCSPACE}[s, c, e]$, the main takeaway of Theorem 1 is that allowing seemingly minor (superconstant) errors in the resetting condition can give an $\mathsf{LCSPACE}$ machine surprising power. A concrete instantiation of this view is the following direct corollary.

▶ **Corollary 2.** *For any $e := e(n)$,*

$$\mathsf{LCL}[e] = \mathsf{CL} \quad \textit{implies} \quad \mathsf{SPACE}[O(e \log n)] \subseteq \mathsf{ZPP}$$

If we revisit the assumption that we cannot hope to prove $\mathsf{SPACE}[e \log n]$ is in ZPP for any $e = \omega(1)$, then Corollary 2 implies the result of [16] is optimal with respect to e; any result of the form $\mathsf{LCL}[\omega(1)] = \mathsf{CL}$ is out of reach.

We also show that our proof extends to catalytic machines with additional power beyond errors, namely *non-deterministic* and *randomized* catalytic space.

▶ **Theorem 3.** *Let $\mathcal{C} \in \{\mathsf{NCSPACE}, \mathsf{BPCSPACE}\}$, and let $s := s(n), c := c(n), e := e(n)$ be such that $e \leq c^{1-\Omega(1)}$. Then*

$$\mathsf{LC}[s, c, e] = \mathcal{C}[\Theta(s + e \log c), \Theta(c)]$$

We briefly remark that the $e \leq c^{1-\Omega(1)}$ restriction in all our results is only needed to get the constant stretch in the catalytic tape; we discuss the unrestricted setting in Section 3.2.

1.4 Open problems

1.4.1 Errors in expectation

A related question asked in [20] is whether or not CL is equivalent to CL with $O(1)$ errors allowed *in expectation* over all starting catalytic tapes. This represents a different notion of distance between catalytic tapes, in opposition to Hamming distance, that may be more applicable to settings such as quantum computation. This question has received some attention in a related form by Bisoyi et al. [1], who introduce *almost* catalytic machines, which perfectly reset some catalytic tapes and are completely unrestricted on others.

However, no general results are known for expected errors – the results in [1] are very structured – and all techniques in our paper fail to restore the tape in pathological cases where a few starting tapes end up with potentially many errors. Furthermore, a barrier result was pointed out by an anonymous reviewer.[2]

[2] The main idea is that allowing virtually unlimited error in an exponentially small fraction of catalytic tapes gives us a strong form of the "compress-or-random" framework of previous papers; we can simulate

1.4.2 Randomized error-prone catalytic space

Recent work of Cook et al. [6] shows that $\mathsf{CSPACE}[s, c] = \mathsf{BPCSPACE}[O(s), \mathrm{poly}(c)]$, which, in conjunction with Theorem 3, seems to indicate that our theorems can be unified to show the connection between ordinary CSPACE and CSPACE which is both randomized and lossy, i.e. $\mathsf{CSPACE}[s + e \log c, c] = \mathsf{LBPCSPACE}[O(s), \mathrm{poly}(c), e]$. This would characterize how deterministic catalytic space handles both natural kinds of "error", namely both error in the output from the randomness and error in resetting the catalytic tape.

However, the proof of [6] only works when $c = 2^{\Theta(s)}$, and our connection to error-prone space incurs an $e \log c$ blowup in free space, putting us outside this regime. A generalization of their result, i.e. showing $\mathsf{CSPACE}[s, c] = \mathsf{BPCSPACE}[O(s), \mathrm{poly}(c)]$ for *every* s and c, would tie off this connection. Note that the polynomial blowup allowed in the catalytic tape means this result would not yield novel derandomization for ordinary space; even for $s, c = O(\log n)$ this would only show that derandomization overheads can be pushed into a polylogarithmic length catalytic tape, which was already shown by Pyne [22].

1.4.3 Lossy catalytic branching programs

Due to the flexibility in the conditions of Theorem 1, the results of Theorem 3 are likely to extend to other settings catalytic settings; for example, it is immediate to extend both results to CSPACE with *advice*. We focus on non-determinism and randomness simply because these are two of the most well-studied catalytic variants, and future works are free to adapt these proofs to their own settings.

In terms of notable omissions, however, one setting where one direction does not yet extend, and which is very related to advice, is the *catalytic branching program* model, which is a syntactic, and by extension non-uniform, way of capturing CSPACE. The issue here is simply that such machines can read and write their entire work tape in one step, which our simulation of CSPACE by $\mathsf{LCSPACE}$ is unequipped to handle. As we discuss in the full version of this paper, showing such branching programs are *reversible* would be sufficient to close this off.

1.4.4 Exact simulation space requirements

In the current simulation of errors using clean space, we use $4e \log c$ clean space. By contrast, in our simulation of clean space using errors, we use only e more errors. If errors can be simulated in clean space $e \log c$ instead, then there is only very low overhead in switching between the two perspectives. This would tighten the correspondence between errors and space that we establish. However, since the distance between two codewords required to correct e errors is $2e + 1$, a different error correction code would be necessary to reach clean space $e \log c$.

---

a randomized algorithm using the catalytic tape as our source of randomness, and in the exponentially unlikely event the tape is not sufficiently entropic we simply erase it and run brute force. Formalizing this intuition and combining it with the results of [6] gives a derandomization barrier to showing even $\mathsf{LCL}[O(1)] = \mathsf{CL}$, namely that randomized TC^1, which is not known to even be in P, would reduce to the lossy code problem.

2 Preliminaries

We begin by defining catalytic machines as introduced by Buhrman et al. [4].

▶ **Definition 4** (Catalytic space). *A catalytic Turing Machine is a space-bounded Turing machine with two work tapes: 1) a read-write work tape of length $s(n)$ which is initialized to $0^{s(n)}$, and 2) a read-write catalytic tape of length $c(n) \leq 2^{s(n)}$ which is initialized to an arbitrary state $\tau \in \{0,1\}^{c(n)}$. On any input $x \in \{0,1\}^n$ and initial catalytic state τ, a catalytic Turing machine has the property that at the end of the computation on input x, the catalytic tape will be in the initial state τ.*

In this work we focus on a relaxation of catalytic space by Gupta, Jain, and Sharma [16], where we are allowed to make some errors in resetting the catalytic tape.

▶ **Definition 5** (Lossy catalytic space). *A lossy catalytic Turing Machine with $e(n)$ errors is a catalytic machine where at the end of the computation on any input $x \in \{0,1\}^n$ and initial catalytic state τ, instead of requiring that the catalytic tape be in state τ, the catalytic tape can be in any state τ' such that τ and τ' differ in at most $e(n)$ locations.*

Lastly we specify the basic complexity classes arising from our two catalytic definitions, as well as their specification to the "logspace" setting, where most research interest at the moment lies.

▶ **Definition 6.** *We write*

- $\mathsf{CSPACE}[s, c]$*: the class of languages which can be recognized by catalytic Turing Machines with work space $s := s(n)$ and catalytic space $c := c(n)$.*
- $\mathsf{LCSPACE}[s, c, e]$*: the class of languages which can be recognized by lossy catalytic Turing Machines with work space $s := s(n)$, catalytic space $c := c(n)$, and $e := e(n)$ errors.*

We additionally write

- $\mathsf{CL} := \mathsf{CSPACE}[O(\log n), \operatorname{poly} n]$
- $\mathsf{LCL}[e] := \mathsf{LCSPACE}[O(\log n), \operatorname{poly} n, e]$

We note that throughout this paper we write $\mathcal{C}[O(f(n))]$ as a shorthand for $\bigcup_{c\in\mathbb{N}} \mathcal{C}[c \cdot f(n)]$ for complexity class $\mathcal{C}$ and function $f(n)$, and similarly for classes with multiple parameters or other asymptotics (e.g. $\mathsf{CSPACE}[\Theta(s), \Theta(c)]$).

3 Main theorem

In this section we will prove Theorem 1. We will do so via a simulation argument for each direction in turn.

3.1 Simulating errors with space

First, we show that $\mathsf{LCSPACE}[s, c, e] \subseteq \mathsf{CSPACE}[O(s + e \log c), O(c)]$. In fact, we will not need any increase in the length of our catalytic tape.

▶ **Theorem 7.** *Let $s := s(n), c := c(n), e := e(n)$. Then*

$$\mathsf{LCSPACE}[s, c, e] \subseteq \mathsf{CSPACE}[s + O(e \log c), c]$$

We note that this was also proven in [16] for the case of $\mathsf{LCL}[O(1)]$, but we will pursue a different proof, based on error-correcting codes, which will allow us to generalize to other catalytic models in Section 4.

Proof. Let M_e be an $\mathsf{LCSPACE}[s, c, e]$ machine. We will devise a $\mathsf{CSPACE}[s + O(e \log c), c]$ machine M_0 which simulates M_e. Note that in this section, we will not use our parameter restriction on e; this direction holds for every setting of s, c, and e. We will presume that $e \leq \frac{c}{\log(c)}$, as the inclusion becomes trivial otherwise.

Our simulation will go via an error-correcting code. In particular we will use *BCH codes*[3] (BCH), named after Bose, Ray-Chaudhuri, and Hocquenghem [3, 17], which we define as per [13, 12]. We define the mapping BCH and prove the following lemma in the appendix to the the full version of our paper.

▶ **Lemma 8.** *Let $q := 2^{\lceil \log(c+e) \rceil}$. There exists a mapping $\mathsf{BCH} : \mathbb{F}_q^q \to \mathbb{F}_q^q$ with the following operations:*

- ***Encoding:*** $\mathsf{Enc}_{\mathsf{BCH}}$ *takes as input a string S of length c, plus an additional $(2e+1)\lceil \log(c+e) \rceil$ bits initialized in 0, and outputs a codeword S_{enc}:*

  $$S + [0]_{(2e+1)\lceil \log(c+e) \rceil} \to_{\mathsf{Enc}} S_{enc}$$

  Furthermore, all outputs S_{enc} generated this way have minimum distance $\delta := 2e+1$ from one another.
- ***Decoding:*** $\mathsf{Dec}_{\mathsf{BCH}}$ *takes as input a string S'_{enc} of length $c + (2e+1)\log(c+e)$, with the promise that there exists a string S of length c such that $\mathsf{Enc}_{\mathsf{BCH}}(S + [0]_{2e \log(c+e)})$ differs from S'_{enc} in at most $\delta/2 - 1 = e$ locations, and outputs this string S:*

  $$S'_{enc} \to_{\mathsf{Dec}} S + [0]_{(2e+1)\log(c+e)}$$

Furthermore, both $\mathsf{Enc}_{\mathsf{BCH}}$ and $\mathsf{Dec}_{\mathsf{BCH}}$ are in place replacements of the input strings, they require at most an additional $O(e \log c)$ free space of memory.

We now move on to the simulation of our $\mathsf{LCSPACE}[s, c, e]$ machine M_e. Our $\mathsf{CSPACE}[s + O(e \log c), c]$ machine M_0 acts as follows:

1. **Initialization:** use the function $\mathsf{Enc}_{\mathsf{BCH}}$ to encode the initial state τ of the catalytic tape into a codeword, using $(2e+1)\lceil \log(c+e) \rceil$ additional bits from clean space,

   $$\tau + [0]_{(2e+1)\lceil \log(c+e) \rceil} \to_{\mathsf{Enc}} \tau_{enc}.$$

2. **Simulation:** Run M_e using clean space s and the first c bits of τ_{enc} as the catalytic tape. When M_e finishes the calculation, we record the answer in a bit of the free work tape. The catalytic tape is, at this point, in a state τ'_{enc} which differs in at most e locations from τ_{enc}.
3. **Cleanup:** use the function $\mathsf{Dec}_{\mathsf{BCH}}$ to detect and correct our resulting catalytic tape τ'_{enc}:

   $$\tau'_{enc} \to_{\mathsf{Dec}} \tau + [0]_{(2e+1)\lceil \log(c+e) \rceil}$$

   Once we finish this process, we output our saved answer and halt.

The correctness of M_0 is clear, as it gives the same output as M_e. By our error guarantee on M_e and the correctness of Dec, our catalytic tape is successfully reset to τ. Our catalytic memory is c as before, while for our free work space we require s bits to simulate M_e, an additional $(2e+1)\lceil \log(c+e) \rceil = (2 + o(1))e \log c$ zero bits for our codewords, and $O(e \log c)$ space for $\mathsf{Enc}_{\mathsf{BCH}}$ and $\mathsf{Dec}_{\mathsf{BCH}}$, for $s + O(e \log c)$ space in total. ◀

[3] Technically because of our parameters, they can even be considered Reed-Solomon codes, which are a special case of BCH codes; nevertheless we follow the presentation of the more general code form.

▶ Note 9. There is an alternative proof of this point, one which gets better parameters and relies on an interesting characterization of space, namely the *reversibility* of space. This proof is a simplification and extension of the one originally provided in [16], and we provide it in the appendix to the full version of our paper for those interested.

3.2 Simulating space with errors

We now show the other direction of Theorem 1, i.e. $\mathsf{CSPACE}[s + e\log c, c]$ is contained in $\mathsf{LCSPACE}[O(s), O(c), O(e)]$.

▶ **Theorem 10.** *Let $s := s(n), c := c(n), e := e(n)$, and $\epsilon > 0$ be such that $e = o(c^{\epsilon/(1+\epsilon)})$. Then*

$$\mathsf{CSPACE}[s + e\log c, c] \subseteq \mathsf{LCSPACE}[s + \log c, (1 + o(1))c, (1 + \epsilon)e]$$

Since $s \geq \log c$ by the definition of a catalytic machine, this achieves the reverse direction of Theorem 1 with very small blowups in s and c, and for e bounded by a small polynomial in c we get a negligible error blowup as well. Note that we allow $\epsilon > 1$, and so our proof is not limited to $e < c^{1/2}$; however, we will pay for larger values of e in the error blowup, and for $e = c^{1-o(1)}$ this factor becomes superconstant.

To understand our construction, we will first prove a version with looser space parameters. This result is incomparable to Theorem 10; although we lose a factor of e in our catalytic space, in exchange we have no restrictions on e and no loss in e either.

▶ **Theorem 11.** *Let $s := s(n), c := c(n), e := e(n)$ be such that c is a power of 2. Then*

$$\mathsf{CSPACE}[s + e\log c, c] \subseteq \mathsf{LCSPACE}[s + (\log e + \log c + 2), (e + 1)c, e]$$

Proof. Let M_0 be a $\mathsf{CSPACE}[s + e\log c, c]$ machine. We will devise a $\mathsf{LCSPACE}[s + (\log e + \log c + 2), (e+1)c, e]$ machine M_e which simulates M_0. By the definition of a Turing machine, we will assume that in any time step M_0 only reads and writes at most one bit on the work tape.

Throughout this proof, we will associate $[2^k]$ with $\{0,1\}^k$ in the usual manner, i.e. subtracting 1 and taking the binary representation, and so we will use them interchangeably. Our workhorse is the following folklore[4] construction:

▶ **Lemma 12.** *For every k, there exists a mapping $\mathsf{mem} : \{0,1\}^{2^k} \to \{0,1\}^k$, computable in space $k + 1$, such that the following holds: for any $\tau \in \{0,1\}^{2^k}$ and any $y \in \{0,1\}^k$,*

$$\mathsf{mem}(\tau \oplus \sigma_y) = \mathsf{mem}(\tau) \oplus y$$

where σ_y is the vector of length 2^k with a single 1 in position y.

Intuitively, Lemma 12 gives us an easily computable mapping where any transformation of the k-bit output string can be achieved by flipping one bit of the 2^k-bit input string, with the location of this single bitflip being determined only by the locations where the current and target output strings differ.[5]

[4] This construction is based on the solution to the so-called "almost impossible chessboard puzzle"; interested readers can find the setup and solution in videos on the YouTube channels 3Blue1Brown (`https://www.youtube.com/watch?v=wTJI_WuZSwE`) and Stand-up Maths (`https://www.youtube.com/watch?v=as7Gkm7Y7h4`). It can also be seen as the syndrome of the Hamming code.

[5] It is not particularly important that the location to flip in τ to achieve $\mathsf{mem}(\tau) \oplus y$ is exactly σ_y, but only that there exists *some* such place to flip, depending only on the mask y, and that this location can be easily calculated given y.

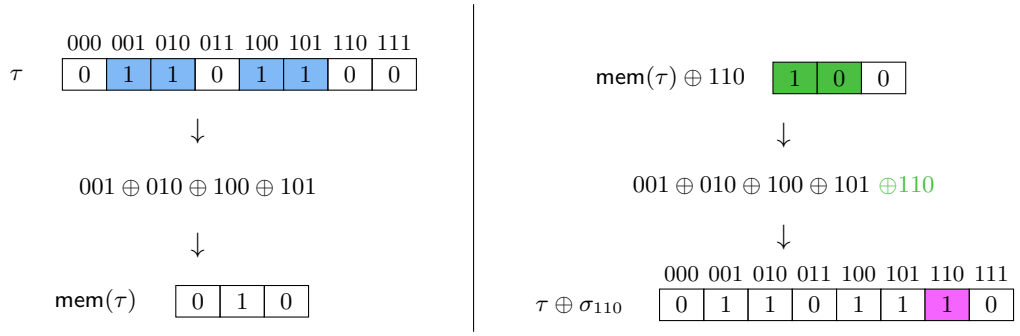


Figure 1 Example of our construction in Lemma 12 for $k = 3$ and $\tau = 01101100$: 1) calculating $\mathsf{mem}(\tau)$ based on the positions of the 1s in τ (blue); 2) how flipping one bit of τ (magenta) allows us to change $\mathsf{mem}(\tau)$ (changes in green).

Proof of Lemma 12. Let $\tau \in \{0,1\}^{2^k}$ be indexed by bitstrings $z \in \{0,1\}^k$. We will define our mapping mem as the entrywise sum of all indices z where $\tau_z = 1$, i.e.

$$\mathsf{mem}(\tau)_j = \bigoplus_{\substack{z \in \{0,1\}^k \\ z_j = 1}} \tau_z$$

This is clearly computable in space $k+1$, as we need only store z and our current sum. Now note that for any y, flipping the entry τ_y, i.e. $\tau \oplus \sigma_y$, flips every $\mathsf{mem}(\tau)_j$ entry where $y_j = 1$ and leaves all other $\mathsf{mem}(\tau)_j$ entry unchanged, which gives $\mathsf{mem}(\tau) \oplus y$ as claimed. ◀

At a high level, our $\mathsf{LCSPACE}[s + (\log e + \log c + 2), (e+1)c, e]$ machine M_e works as follows. M_e will use its s bits of free memory and the first c bits of catalytic memory to represent their equivalent blocks in M_0, i.e. s bits of free memory and c bits of catalytic memory. We will break the remaining $e \cdot c$ bits of our catalytic tape of M_e into e blocks $B_1 \dots B_e$ of size c each, and we denote by τ_i the memory in block B_i. We apply Lemma 12 with $k = \log c$ – recall that we assume c is a power of 2 – and use each $\mathsf{mem}(\tau_i)$ to represent an additional $\log c$ bits of free memory, giving us an additional $e \log c$ bits of memory in total. Our additional workplace memory will be used to compute the mapping, serve as pointers, and other assorted tasks.

Before formally stating M_e, we mention a special case of the construction of Lemma 12, which will allow us to use it in a reversible operational manner.

▷ Claim 13. Let $\tau_0, \tau_1, \tau_2 \dots \tau_{t-1}, \tau_t \in \{0,1\}^{2^k}$ be such that 1) τ_i and τ_{i-1} differ in exactly one coordinate for all $i \in [t]$; 2) $\mathsf{mem}(\tau_i)$ and $\mathsf{mem}(\tau_{i-1})$ differ in exactly one coordinate for all $i \in [t]$; and 3) $\mathsf{mem}(\tau_t) = \mathsf{mem}(\tau_0)$. Then $\tau_t = \tau_0$.

Proof. For all $i \in [t]$, let $b_i \in [k]$ be the location where $\mathsf{mem}(\tau_i)$ and $\mathsf{mem}(\tau_{i-1})$ differ, and let β_i be the location where τ_i and τ_{i-1} differ. Since $\mathsf{mem}(\tau_t) = \mathsf{mem}(\tau_0)$, each value $j \in [k]$ must appear an even number of times in the list $b_1 \dots b_t$, and since flipping any location $j \in [k]$ in $\mathsf{mem}(\tau)$ can only be obtained from one flip in τ at the unique location $2^j \in [2^k]$, it follows that each value $J \in [2^k]$ must appear an even number of times in the list $\beta_1 \dots \beta_t$. This means that τ_t is τ_0 with each bit flipped an even number of times, or in other words $\tau_t = \tau_0$. ◁

We now concretely define our machine M_e:

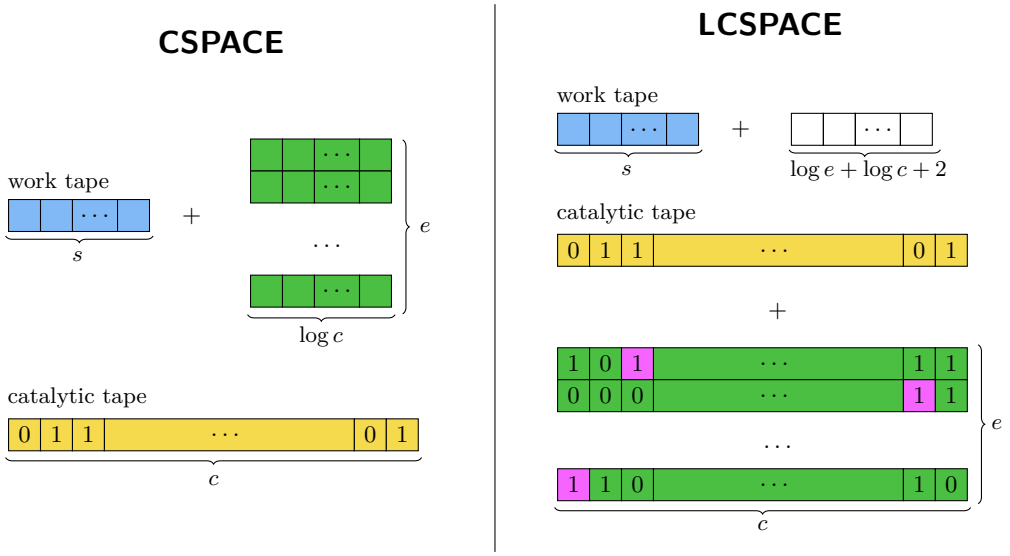


Figure 2 Structure of simulation, where like colors mean space used for same purpose. s bits of work space (blue) and c bits of catalytic space (gold) of the CSPACE machine are directly simulated in the LCSPACE machine, while each $\log c$ length chunk of additional work space of the CSPACE machine (green) is simulated by c catalytic bits and one error (magenta) of the LCSPACE machine. An additional $\log e + \log c + 2$ bits of work memory are used by the LCSPACE machine for scratch work, such as computing mem and addressing into the catalytic memory.

1. **Initialization:** for each block B_i, calculate mem_i and flip the mem_ith element of B_i:

$$\tau_i \to \tau_i \oplus \sigma_{\mathsf{mem}(\tau_i)} \qquad \forall i \in [e]$$

   Define τ_i^{enc} to be the memory after this process. Note that we now have exactly e errors on the tape, one in each τ_i^{enc}, and we are guaranteed that

$$\mathsf{mem}(\tau_i^{enc}) = 0^{\log c} \qquad \forall i \in [e]$$

2. **Simulation:** run M_0 using s free work bits and c catalytic bits, with the concatenation of the values mem_i as the other $e \log c$ free work bits. To do this, whenever we read or write a bit in our $e \log c$ bits of memory, we find the B_i responsible for this bit, calculate mem_i, and update τ_i using one bitflip to reflect how mem_i changes according to the operation of M_0:

$$\tau_i^{enc} \to \tau_i^{enc} \oplus \sigma_{2^j} \qquad \text{update occurs in bit } j \text{ of block } i$$

   If mem_i is unchanged, we make no updates to τ_i.
3. **Cleanup:** when we reach the end of M_0's computation, record the answer on the free work tape and set each mem_i value to $0^{\log c}$ one bit at a time:

$$\tau_i^{enc} \to \tau_i^{enc} \oplus \sigma_{2^j} \qquad \forall i \in [e], j : \mathsf{mem}(\tau_i^{enc}))_j = 1$$

   Once we finish this process, we output our saved answer and halt.

The correctness of M_e is clear, as we output the same value as M_0. Our catalytic space usage is $c + ec$ by construction, while our free space usage is s to simulate M_0, one extra bit to save the output, and any additional space required to handle the simulation of the

additional $e \log c$ work bits. In particular, we need $\log e$ bits for a pointer into B_i and $\log c + 1$ bits for the computation of mem_i by Lemma 12, for a total space usage of $s + \log e + \log c + 2$ as claimed.

We also claim that our lossy catalytic condition is satisfied. By the property of M_0, we make no errors on the c catalytic bits used for the simulation of M_0's catalytic space. The initialization step introduces at most one error per τ_i, thus giving at most e errors on to the catalytic tape. After the initialization step, each other update to τ_i corresponds to changing a single bit in $\mathsf{mem}(\tau_i)$, with the final value being the same as the value after initialization. Thus by Claim 13 we restore the catalytic tape to its position after the initialization phase, and so our end state corresponds to our original catalytic tape with at most e errors, i.e. those induced by the initialization phase, as required. ◀

We now return to Theorem 10, which requires only a small modification of the above proof, namely to break the the catalytic tape into more, smaller blocks, which reduces its required length at the cost of a few extra errors. This modification works because the number of pure bits represented is logarithmic in the length of the block, and so making the blocks smaller barely affects the number of bits represented; for example, $c/2$ bits in a block still lets you represent $\log(c) - 1$ bits, so half the size only loses one bit per block.

Proof of Theorem 10. Let M_0 be a $\mathsf{CSPACE}[s + e \log c, c]$ machine. We will devise a $\mathsf{LCSPACE}[s + e \log c, (1 + o(1))c, (1 + \epsilon)e]$ machine M_e which simulates M_0, where ϵ satisfies $e = o(c^{\epsilon/(1+\epsilon)})$.

An easy manipulation gives us $c = \omega(e^{1+1/\epsilon})$, and so there exists a function $\delta = \omega(1)$ such that $c \geq (\delta e)^{1+1/\epsilon}$. We will have the same approach as Theorem 11, but now we use $(1 + \epsilon)e$ blocks of length $2^{\lceil \log c/(\delta e) \rceil}$ each, i.e., using Lemma 12 for $k = \lceil \log c/(\delta e) \rceil$. Since we introduce one error per block, the number of errors the machine makes is at most $(1 + \epsilon)e$, while our total catalytic tape has length

$$c + (1 + \epsilon)e \cdot 2^{\lceil \log c/(\delta e) \rceil} \leq c + (1 + \epsilon)e \cdot \frac{2c}{\delta e} = c \cdot (1 + o(1))$$

Lastly, we can use our additional catalytic memory to simulate a work tape of length

$$(1 + \epsilon)e \cdot \log 2^{\lceil \log(c/\delta e) \rceil} \geq e \cdot (1 + \epsilon) \log \frac{c}{\delta e}$$

and by manipulating our starting assumption we get that

$$\begin{aligned} c &\geq (\delta e)^{1+1/\epsilon} \\ (c/\delta e)^{\epsilon} &\geq \delta e \\ \epsilon \log(c/\delta e) &\geq \log \delta e \\ \log c - \log \delta e + \epsilon \log(c/\delta e) &\geq \log c \\ e \cdot (1 + \epsilon) \log(c/\delta e) &\geq e \log c \end{aligned}$$

thus giving us a simulation of $e \log c$ work bits as claimed. The correctness and lossy catalytic condition can then be argued as above, and our free space usage is s plus an additional $\log(c \cdot o(1)) + 2 \leq \log c$ bits, for a total space usage of $s + \log c$ as claimed. ◀

4 Further consequences

With this, we have concluded our main theorem and proof. We now move to corollaries and extensions.

4.1 Lossy catalytic logspace with superconstant errors

As stated in the introduction, it immediately follows from Theorem 1 that proving $\mathsf{LCL}[e] = \mathsf{CL}$ is likely difficult, if not false, for superconstant values of e.

Proof of Corollary 2. This follows immediately from the fact that

$$\begin{aligned}\mathsf{LCSPACE}[O(\log n), \operatorname{poly} n, e] &= \mathsf{CSPACE}[O(\log n + e \log(\operatorname{poly} n)), \operatorname{poly} n] \\ &= \mathsf{CSPACE}[O(e \log n), \operatorname{poly} n] \\ &\supseteq \mathsf{SPACE}[O(e \log n)]\end{aligned}$$

combined with the fact that $\mathsf{CL} \subseteq \mathsf{ZPP}$ by [4]. ◀

4.2 Lossy catalytic space with other resources

As mentioned in Section 1, there are many extensions of the base catalytic model besides LCSPACE, such as randomized, non-deterministic, and non-uniform CSPACE. So far, however, there has been little discussion of classes where more than one such external resource has been utilized. In this section, we will discuss two of the aforementioned models – randomized and non-deterministic CSPACE – in the presence of errors.

▶ **Definition 14.** *Let f be a Boolean function on n inputs.*

- *A non-deterministic catalytic Turing machine for f is a catalytic machine M which, in addition to its usual input x, has access to a read-once witness string w, of length at most exponential in the total space of M, based on x, which has the following properties:*
  - *if $f(x) = 1$, then there exists a witness w such that $M(x, w) = 1$*
  - *if $f(x) = 0$, then for every witness string w, $M(x, w) = 0$*

  Furthermore, for every witness string w, $M(x, w)$ restores the catalytic tape to its original configuration.
- *A randomized catalytic Turing machine for f is a catalytic machine M which, in addition to its usual input x, has access to a read-once uniformly random string r, of length at most exponential in the total space of M, such that*

$$\mathbf{Pr}_r[M(x, r) = f(x)] \geq 2/3.$$

  Furthermore, for every witness string w, $M(x, w)$ restores the catalytic tape to its original configuration.

We write

- $\mathsf{NCSPACE}[s, c]$*: the class of languages which can be recognized by non-deterministic catalytic Turing Machines with work space $s := s(n)$ and catalytic space $c := c(n)$.*
- $\mathsf{BPCSPACE}[s, c]$*: the class of languages which can be recognized by randomized catalytic Turing Machines with work space $s := s(n)$ and catalytic space $c := c(n)$.*

Furthermore, for $\mathcal{C} \in \{\mathsf{NCSPACE}, \mathsf{BPCSPACE}\}$, we define $\mathsf{LC}[s, c, e]$ to be the class $\mathcal{C}[s, c]$ with the catalytic resetting definition replaced by the LCSPACE *resetting definition, i.e., where e errors are allowed to remain on the catalytic tape at the end of any computation.*

Note that we allow the errors to depend on the witness and randomness, respectively. Without delving too deep into these models, however, it is clear that our proof of Theorem 1 carries over to all the above definitions.

Proof sketch of Theorem 3. As earlier, we need to show both directions. We will prove the same two equivalences as in Theorems 7 and 10, namely

1. $\mathsf{L}\mathcal{C}[s, c, e] \subseteq \mathcal{C}[s + O(e \log c), c]$
2. $\mathcal{C}[s + e \log c, c] \subseteq \mathsf{L}\mathcal{C}[s + \log c, (1 + o(1))c, (1 + \epsilon)e]$

Both directions will follow immediately via the same simulation as before. For the forward direction, we simulate our $\mathsf{L}\mathcal{C}$ machine by a $\mathcal{C}$ machine as usual, and then at the last step we correct the changes using our BCH code as before; this works just as before because the code allow us to correct any e errors on the catalytic tape, regardless of how they came about.

For the reverse direction, again our $\mathcal{C}$ machine will only read or write one bit per time step, and so we use the same mem_i approach to simulating our additional $e \log c$ bits of free memory, which does not change based on the operation of the rest of the machine. As in the forward direction, simulating the actual workings of the $\mathcal{C}$ machine via the $\mathsf{L}\mathcal{C}$ machine, given our method of simulating the $e \log c$ additional bits of memory, is straightforward, and our resetting step at the end again resets our extra catalytic tape regardless of the computation path the $\mathcal{C}$ machine takes. ◀

References

1. Sagar Bisoyi, Krishnamoorthy Dinesh, Bhyabya Deep Rai, and Jayalal Sarma. Almost-catalytic and property-catalytic computation, 2024.
2. Sagar Bisoyi, Krishnamoorthy Dinesh, and Jayalal Sarma. On pure space vs catalytic space. *Theoretical Computer Science (TCS)*, 921:112–126, 2022. doi:10.1016/J.TCS.2022.04.005.
3. Raj Chandra Bose and Dwijendra K Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and control*, 3(1):68–79, 1960. doi:10.1016/S0019-9958(60)90287-4.
4. Harry Buhrman, Richard Cleve, Michal Koucký, Bruno Loff, and Florian Speelman. Computing with a full memory: catalytic space. In *ACM Symposium on Theory of Computing (STOC)*, pages 857–866, 2014. doi:10.1145/2591796.2591874.
5. Harry Buhrman, Michal Koucký, Bruno Loff, and Florian Speelman. Catalytic space: Non-determinism and hierarchy. *Theory of Computing Systems (TOCS)*, 62(1):116–135, 2018. doi:10.1007/S00224-017-9784-7.
6. James Cook, Jiatu Li, Ian Mertz, and Edward Pyne. The structure of catalytic space: Capturing randomness and time via compression. *Electronic Colloquium on Computational Complexity (ECCC)*, TR24-106, 2024. URL: https://eccc.weizmann.ac.il/report/2024/106.
7. James Cook and Ian Mertz. Catalytic approaches to the tree evaluation problem. In *ACM Symposium on Theory of Computing (STOC)*, pages 752–760. ACM, 2020. doi:10.1145/3357713.3384316.
8. James Cook and Ian Mertz. Encodings and the tree evaluation problem. *Electronic Colloquium on Computational Complexity (ECCC)*, TR21-054, 2021. URL: https://eccc.weizmann.ac.il/report/2021/054.
9. James Cook and Ian Mertz. Trading time and space in catalytic branching programs. In *IEEE Conference on Computational Complexity (CCC)*, volume 234 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:21, 2022. doi:10.4230/LIPIcs.CCC.2022.8.
10. James Cook and Ian Mertz. Tree evaluation is in space O(log n · log log n). In *ACM Symposium on Theory of Computing (STOC)*, pages 1268–1278. ACM, 2024. doi:10.1145/3618260.3649664.
11. Samir Datta, Chetan Gupta, Rahul Jain, Vimal Raj Sharma, and Raghunath Tewari. Randomized and symmetric catalytic computation. In *CSR*, volume 12159 of *Lecture Notes in Computer Science (LNCS)*, pages 211–223. Springer, 2020. doi:10.1007/978-3-030-50026-9_15.

12 Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Syndrome encoding and decoding of bch codes in sublinear time, 2006. URL: https://www.cs.bu.edu/~reyzin/code/bch-excerpt.pdf.

13 Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Advances in Cryptology - EUROCRYPT 2004*, 2004.

14 Dean Doron, Edward Pyne, and Roei Tell. Opening up the distinguisher: A hardness to randomness approach for BPL = L that uses properties of BPL. In *Proceedings of the 56nd Annual ACM Symposium on Theory of Computing, STOC 2024*, pages 2039–2049, 2024. doi:10.1145/3618260.3649772.

15 Chetan Gupta, Rahul Jain, Vimal Raj Sharma, and Raghunath Tewari. Unambiguous catalytic computation. In *Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 150 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.FSTTCS.2019.16.

16 Chetan Gupta, Rahul Jain, Vimal Raj Sharma, and Raghunath Tewari. Lossy catalytic computation. *Computing Research Repository (CoRR)*, abs/2408.14670, 2024. doi:10.48550/arXiv.2408.14670.

17 Alexis Hocquenghem. Codes correcteurs d'erreurs. *Chiffers*, 2:147–156, 1959.

18 Michal Koucký. Catalytic computation. *Bulletin of the EATCS (B.EATCS)*, 118, 2016. URL: http://eatcs.org/beatcs/index.php/beatcs/article/view/400.

19 Jiatu Li, Edward Pyne, and Roei Tell. Distinguishing, predicting, and certifying: On the long reach of partial notions of pseudorandomness. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, to appear, 2024.

20 Ian Mertz. Reusing space: Techniques and open problems. *Bulletin of the EATCS (B.EATCS)*, 141:57–106, 2023. URL: http://eatcs.org/beatcs/index.php/beatcs/article/view/780.

21 Aaron Potechin. A note on amortized branching program complexity. In *IEEE Conference on Computational Complexity (CCC)*, volume 79 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.CCC.2017.4.

22 Edward Pyne. Derandomizing logspace with a small shared hard drive. In *39th Computational Complexity Conference, CCC 2024*, volume 300 of *LIPIcs*, pages 4:1–4:20, 2024. doi:10.4230/LIPICS.CCC.2024.4.

23 Robert Robere and Jeroen Zuiddam. Amortized circuit complexity, formal complexity measures, and catalytic algorithms. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 759–769. IEEE, 2021. doi:10.1109/FOCS52979.2021.00079.

Parameterized Geometric Graph Modification with Disk Scaling

Fedor V. Fomin
University of Bergen, Norway

Petr A. Golovach
University of Bergen, Norway

Tanmay Inamdar
Indian Institute of Technology Jodhpur,India

Saket Saurabh
The Institute of Mathematical Sciences, HBNI, Chennai, India
University of Bergen, Norway

Meirav Zehavi
Ben-Gurion University of the Negev, Beer-Sheva, Israel

Abstract

The parameterized analysis of graph modification problems represents the most extensively studied area within Parameterized Complexity. Given a graph G and an integer $k \in \mathbb{N}$ as input, the goal is to determine whether we can perform at most k operations on G to transform it into a graph belonging to a specified graph class $\mathcal{F}$. Typical operations are combinatorial and include vertex deletions and edge deletions, insertions, and contractions. However, in many real-world scenarios, when the input graph is constrained to be a geometric intersection graph, the modification of the graph is influenced by changes in the geometric properties of the underlying objects themselves, rather than by combinatorial modifications. It raises the question of whether vertex deletions or adjacency modifications are necessarily the *most appropriate* modification operations for studying modifications of geometric graphs.

We propose the study of the disk intersection graph modification through the scaling of disks. This operation is typical in the realm of topology control but has not yet been explored in the context of Parameterized Complexity. We design parameterized algorithms and kernels for modifying to the most basic graph classes: edgeless, connected, and acyclic. Our technical contributions encompass a novel combination of linear programming, branching, and kernelization techniques, along with a fresh application of bidimensionality theory to analyze the area covered by disks, which may have broader applicability.

2012 ACM Subject Classification Theory of computation → Packing and covering problems; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases parameterized algorithms, kernelization, spreading points, distant representatives, unit disk packing

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.51

Related Version *Full Version*: `https://arxiv.org/abs/2411.13171`

Funding *Fedor V. Fomin*: Supported by the Research Council of Norway via the project BWCA (grant no. 314528).
Petr A. Golovach: Supported by the Research Council of Norway via the project BWCA (grant no. 314528).
Tanmay Inamdar: Supported by IITJ Research Initiation Grant (grant number I/RIG/TNI/20240072) and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme.
Saket Saurabh: Supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819416); and Swarnajayanti Fellowship grant DST/SJF/MSA-01/2017-18.



© Fedor V. Fomin, Petr A. Golovach, Tanmay Inamdar, Saket Saurabh, and Meirav Zehavi;
licensed under Creative Commons License CC-BY 4.0



Meirav Zehavi: Supported by the European Research Council (ERC) grant no. 101039913 (PARAPATH).

Acknowledgements We thank the anonymous reviewers for their valuable feedback.

1 Introduction

In a *graph modification problem*, the input consists of an n-vertex graph G and an integer k. The objective is to determine whether k *modification operations* – such as vertex deletions, or edge deletions, insertions or contractions, or combinations thereof – are sufficient to obtain a graph with prescribed structural properties such as being planar, bipartite, chordal, interval, acyclic or edgeless. Graph modification problems encompass some of the fundamental challenges in graph theory and graph algorithms. They represent the most intensively studied area of research within Parameterized Complexity [23, 20, 32, 47, 8, 13, 14, 9, 2, 36, 3, 27, 11, 38, 37] (this list is illustrative rather than comprehensive; see [19, 23, 20, 32] for more information). Here, the number of allowed modifications, k, is considered a *parameter*. With respect to k, we seek a *fixed parameter tractable* (FPT) algorithm, namely, an algorithm whose running time has the form $f(k)n^{\mathcal{O}(1)}$ for some computable function f. It would not be an exaggeration to state that the design of FPT algorithms for graph modification problems has played one of the most central roles in the development of Parameterized Complexity as a field.

In recent years, parametrized analysis of classic graph problems restricted to geometric intersection graphs – that is, where the input graph G belongs to a prespecified class of geometric intersection graphs – has gained a lot of interest. Naturally, particular attention has been given to graph modification problems restricted to geometric intersection graphs. For example, we refer to [4, 41, 42, 33, 35, 31, 21, 29, 45, 28, 22, 10, 30]. While most classic NP-hard graph problems become polynomial-time solvable on proper interval, interval and circular arc graphs, they remain NP-hard on most other geometric intersection graph classes of interest, which gives rise to their parameterized analysis (see the aforementioned papers). Geometric intersection graph classes studied in this context include, for example, unit disk and disk graphs, unit square and square graphs, convex polygon and polygon graphs, map graphs, string graphs, and the generalization of these classes to higher dimensions. However, one central question arises:

> Are vertex deletion and edge deletion/insertion/contraction the *right* modification operations to use when dealing with graph modification problems on geometric intersection graphs?

Geometric intersection graphs naturally give rise to modification operations that are *geometric* – specifically, we modify the graph by modifying its geometric structure! Arguably, the most fundamental operations in this regard are *scaling* and *shifting (i.e., moving)* the input geometric objects.

> In this paper, our primary objective is to launch a systematic investigation into parameterized graph modification problems concerning geometric intersection graphs, utilizing the operation of scaling the input geometric structure. We regard a significant aspect of our contribution as both conceptual and foundational, with the goal of paving the way for this emerging research direction.

1.1 Topology Control and Disk Scaling as a Modification Operation

Topology Control is a fundamental technique used in wireless ad hoc network and sensor networks to reduce energy consumption and radio interference [46]. This technique is used to control the topology of the graph representing the communication links between network nodes. The usual goal is to maintain or achieve some global graph property (like being connected), or reduce interference (i.e., obtain an edgeless graph) while reducing energy consumption. The geometric operation of scaling – i.e., shrinking or expanding radii – of disks has received significant attention within research of wireless network design due to its relevance to power consumption and interference in the networks. From a theoretical perspective, there is also a large body of literature on algorithmic methods for range assignment in wireless sensor and ad-hoc networks, see [39, 17, 12, 15, 48] for a (very incomplete) sample of various approaches, and it is also of interest for other purposes (see, e.g., [1, 24, 7]). Due to this significance, we consider problems that use disk scaling (i.e., shrinking or expanding a subset of disks) in order to achieve certain topological properties in graph. Our work in fact fits in the larger framework of geometric graph modification, which has been a growing trend in recent years.

Recent work on geometric graph modification. Recently, Fomin et al. [26, 25] have studied some problems related to (re)packing of unit disks and point spreading from the perspective of parameterized complexity. Taking a broader perspective, these problems can also be seen as working in the model of geometric graph modification via *geometric operations*, also studied in [17, 5, 16]. Indeed, these papers consider *movement* of points or disks as an operation to achieve intersection graph that is edgeless. In this paper, we focus on *scaling* (i.e., shrinking or expanding) as the operation, and in addition to edgelessness, we also consider acyclicity and connectivity. Another common theme with the work of [25] is that they also combine enumeration of partial solutions from the (partial) kernel, and then reduce the problem to checking a system of polynomial inequalities. This high level scheme is similar to the one described for designing FPT algorithms for Min k-Shrinking to Independence/Acyclicity, where we use an LP. But this similarity is only superficial, and seems unavoidable due to the inherently *continuous nature* of the problems. The specific context, and in particular, the kind of arguments used to arrive at a point where we can use LP, are quite involved, and different from that in [25], as we will explain this further in the following section. Nevertheless, we hope that using geometric operations for graph modification opens up a new research direction, and we return to this topic in the conclusion. Next, we formally introduce our model and problem definitions, and subsequently delve into the overview of our results.

1.2 Problem Definitions

We fix some notation. Let P be a finite family of points in the plane, and $r : P \to \mathbb{R}_{\geq 0}$ be an assignment of points to radii. Then, the disk graph associated with P and R is defined as follows. Let $\mathcal{D}(P, r)$ be a set that contains the disk centered at p with radius $r(p)$ for each $p \in P$, and let $G(P, r)$ denote the disk graph corresponding to $\mathcal{D}(P, r)$, i.e., a graph where we have a vertex for each point in P, and we have an edge between two points if and only if their corresponding disks intersect. Let $\mathbf{1} : P \to \{1\}$ denote the unit radius assignment, and let $G(P, \mathbf{1})$ denote the *unit disk graph* corresponding to a set of points P.

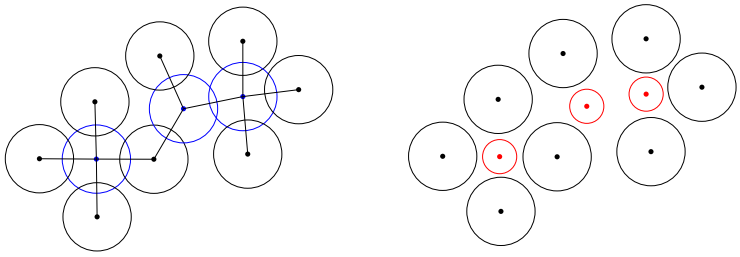


Figure 1 Left: Original intersection graph $G = G(P, \mathbf{1})$ denoting an instance $(P, 1/2, 3)$ of k-Shrinking to Independence. Right: Edgeless graph $G(P, r)$ that shrinks three disks to radius $1/2$ (shown in red).

1.2.1 Shrinking to Achieve Edgelessness and Acyclicity

The first set of problems is related to shrinking a subset of disks in order to achieve two fundamental graph properties, namely, that of being edgeless and acyclic. The general setting is as follows. We are given a unit disk graph $G(P, \mathbf{1})$ corresponding to a set of n points $P \subset \mathbb{R}^2$. We want to *shrink* the disks corresponding to a subset $S \subseteq P$ (i.e., change the radii of disks of each point in S to a value smaller than 1), such that the resulting intersection graph induces an independent set (i.e., is edgeless). We consider two variants of this setting. First, we have a cardinality constrained problem, called k-Shrinking to Independence (see Figure 1).

k-Shrinking to Independence
Input: A finite set of points $P \subset \mathbb{R}^2$, a constant $0 \leq \alpha \leq 1$, and an integer $k \geq 0$.
Task: Decide whether there exists a solution (S, r) such that
1. $S \subseteq P$ is a set of size **at most** k corresponding to shrunken disks,
2. corresponding radii $r : P \to \mathbb{R}_{\geq 0}$ such that $r(p) = \begin{cases} 1 & \text{if } p \notin S \\ \alpha & \text{if } p \in S \end{cases}$
3. $G(P, r)$ is edgeless

We also consider a cost-minimization version of the problem, called Min k-Shrinking to Independence, where the cost of a solution is the total change in the radii, or the sum of *shrinking factors*.

Min k-Shrinking to Independence
Input: A finite set of points $P \subset \mathbb{R}^2$, a constant $0 \leq \alpha \leq 1$, an integer $k \geq 0$, and budget $\mu \geq 0$.
Task: Decide whether there exists a solution (S, r) such that
1. $S \subseteq P$ is a set of size **at most** k corresponding to shrunken disks,
2. corresponding radii $r : P \to \mathbb{R}_{\geq 0}$ where $\begin{cases} r(p) = 1 & \text{if } p \notin S \\ \alpha \leq r(p) \leq 1 & \text{if } p \in S \end{cases}$
3. $G(P, r)$ is edgeless, and
4. $\mathsf{cost}(S, r) := \sum_{p \in S} (1 - r(p)) \leq \mu$.

Analogous to k-Shrinking to Independence, we define k-Shrinking to Acyclicity, where the goal is to achieve $G(P, r)$ that is *acyclic* instead of edgeless (as in item 3 in the definition) (see Figure 2 for an example). Min k-Shrinking to Acyclicity is the

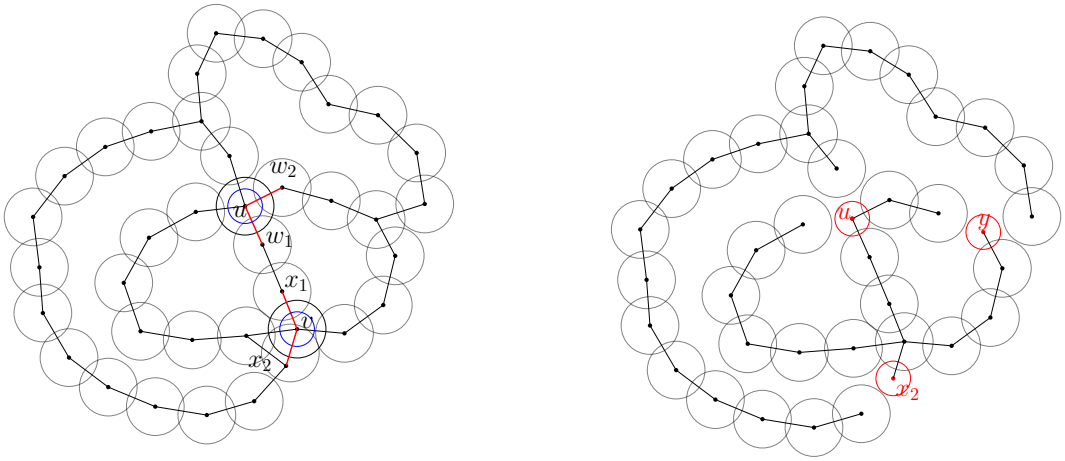


Figure 2 Left: Original intersection graph $G = G(P, \mathbf{1})$ corresponding to an instance $(P, 1/2, 3)$ of k-Shrinking to Acyclicity. Note that $\{u, v\}$ is a minimum feedback vertex set for G. However, even after shrinking the corresponding disks to $1/2$ (shown in blue), the edges $\{uw_1, uw_2, vx_1, v_2\}$ (shown in red) are present in the resulting intersection graph, showing that shrinking disks is not equivalent to vertex deletion. Right: A solution that shrinks three disks (shown in red) to radius $1/2$ corresponding to u, x_2, y, resulting in an acyclic graph $G(P, r)$.

"cost-minimization" version (similar to Min k-Shrinking to Independence), where we want to determine whether we can shrink at most k disks to radii in $[\alpha, 1]$ in order to achieve an acyclic subgraph, such that $\mathsf{cost}(S, r) \leq \mu$.

Note that the operation of shrinking a disk to radius 0 is not necessarily equivalent to deleting the corresponding vertex. Thus, even when $\alpha = 0$, k-Shrinking to Acyclicity (resp. k-Shrinking to Independence) is not exactly equivalent to finding a feedback vertex set (resp. vertex cover) of size k in the original unit disk graph. Nevertheless, we will make use of this connection between the two problems. Now, we turn to the second class of problems considered in this paper.

1.2.2 Shrinking/Expanding to Connectivity

Connectedness is another fundamental graph property in topology control. We consider two variants: shrinking *at least* k disks while *retaining connectivity*, and expanding *at most* k disks to *achieve connectivity*.[1] Next, we define the two problems.

In the first problem, called k-Shrinking to Connectivity[2], we are given a connected unit disk graph $G(P, \mathbf{1})$ corresponding to a set of n points $P \subset \mathbb{R}^2$. We want to *shrink* the disks corresponding to a subset $S \subseteq P$ of size *at least* k (i.e., change the radii of disks corresponding to S to a value smaller than 1), such that the resulting intersection graph *remains* connected.

[1] Observe that expansion versions do not make sense for the problems considered in the earlier section, where we want to obtain an edgeless/acyclic graph – if the graph is not already edgeless/acyclic, one cannot achieve the desired property by expanding a disk.

[2] Ideally, the problem should be named k-Shrinking *While Retaining* Connectivity. However, for the sake of brevity, we stick with k-Shrinking to Connectivity.

k-Shrinking to Connectivity
Input: A finite set of points $P \subset \mathbb{R}^2$, a constant $0 \leq \alpha \leq 1$, and a parameter $k \geq 0$.
Task: Decide whether there exists a solution (S, r) such that
- $S \subseteq P$ is a set of size **at least** k corresponding to shrunken disks,
- corresponding radii $r : P \to \mathbb{R}_{\geq 0}$ such that $r(p) = \begin{cases} 1 & \text{if } p \notin S \\ \alpha & \text{if } p \in S \end{cases}$
- $G(P, r)$ is connected.

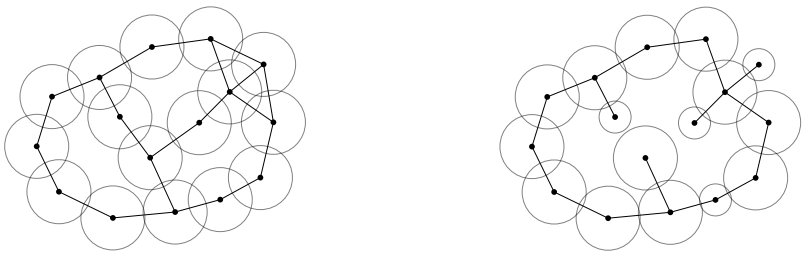


Figure 3 Left: Original intersection graph $G = G(P, \mathbf{1})$ corresponding to an instance $(P, 1/2, 4)$ of k-Shrinking to Connectivity. Right: A solution that shrinks 4 disks to radius $1/2$ while maintaining connectivity in the resulting intersection graph $G(P, r)$.

The next problem, which we call k-Expanding to Connectivity, is in a sense, the "complement version" of k-Shrinking to Independence. Here, we are given a unit disk graph $G(P, \mathbf{1})$ corresponding to a set of n points $P \subseteq \mathbb{R}^2$. We want to *expand* the disks corresponding to a subset $S \subseteq P$ of size at most k (i.e., change the radii of the disks corresponding to S to a value larger than 1), such that the resulting intersection graph *becomes* connected. Formally, the problem is defined as follows.

k-Expanding to Connectivity
Input: A finite set of points $P \subset \mathbb{R}^2$, a constant $\alpha \geq 1$, and a parameter $k \geq 0$.
Task: Decide whether there exists a solution (S, r) such that
- $S \subseteq P$ is a set of size **at most** k corresponding to expanded disks,
- corresponding radii $r : P \to \mathbb{R}_{\geq 0}$ such that $r(p) = \begin{cases} 1 & \text{if } p \notin S \\ \alpha & \text{if } p \in S \end{cases}$
- $G(P, r)$ is connected.

2 Overview of the Technical Results

Our algorithmic technical contribution is, mainly, twofold:

- The first contribution is a novel combination of linear programming, kernelization and branching/exhaustive search: we observe that linear programming can handle continuous domains, kernelization can reduce the search space within these domains, and, when combined with exhaustive search/branching, these can yield the desired solution. While several of our results are based on this approach, we would like to highlight, in particular, the FPT algorithm for k-Shrinking to Acyclicity (see Theorem 8), which exhibits a more intricate combination.

Table 1 A summary of the results in the paper. Additionally, we also show that all of the problems to be NP-hard, which is not mentioned in the table.

Objective	Type of Result	Cardinality version	Cardinality and cost-minimization version
k-Shrinking to Independence	Kernelization	$\mathcal{O}(k)$ partial kernel (Thm. 4), true polynomial kernel (Thm. 5)	$\mathcal{O}(k)$ partial kernel (Thm. 4)
	Parameterized complexity	Subexponential FPT (Thm. 1)	FPT (Thm. 6)
	Approximation	EPTAS (Thm. 10)	Bicriteria EPTAS (Thm. 10)
k-Shrinking to Acyclicity	Kernelization & Compression	True polynomial kernel (Thm. 9)	Partial polynomial compression (Thm. 7)
	Parameterized complexity	Subexponential FPT (Thm. 2)	FPT (Thm. 8)
k-Shrinking to Connectivity	Parameterized complexity	Subexponential FPT (Thm. 3)	–
A generalization of k-Expanding to Connectivity	Parameterized complexity	W[1]-hard (Full version)	–

- The second contribution is, in a sense, the first "geometric" use of bidimensionality theory. Here, the application is tied up with the analysis of the different areas covered by disks. We employ this approach to design a subexponential FPT algorithm for k-Shrinking to Connectivity (see Theorem 3).

We believe that our two approaches are quite general in nature, and will be applicable to other parameterized problems that involve placement of geometric objects in the Euclidean space, particularly when dealing with continuous domains. We summarize our results in Table 1. In this extended abstract, we give an in-depth overview of our algorithmic results, and the technical details are deferred to the full version. One remark is that, all of these problems are shown to be NP-hard, and these NP-hardness results are shown via reduction from well-known planar problems, namely, Independent Set, Feedback Vertex Set in planar graphs, and Planar 3-SAT, respectively. Now we turn to the technical overview of the rest of the results.

2.1 Subexponential FPT Algorithms

In this section, we discuss the first technical contribution of our paper, namely, to design subexponential FPT algorithms for k-Shrinking to Independence, k-Shrinking to Acyclicity and k-Shrinking to Connectivity (specifically the last problem).

Independence and Acyclicity. Recall that in these problems, the goal is to shrink at most k disks to radius α such that the resulting intersection graph is edgeless (resp. acyclic). For the sake of ease of exposition, let us suppose that α, the radius of the shrunken disks, is a constant. We will start by eliminating *large* cliques, i.e., we show using packing arguments that, if the largest clique in the original intersection graph $G = G(P, \mathbf{1})$ has size $\Omega_\alpha(1)$ [3], then no solution can shrink any subset of disks to radius α and remove all the edges. This bounds

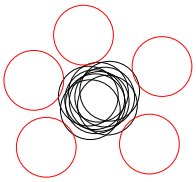
[3] For the sake of simplicity, we use α in the subscript of $\mathcal{O}(\cdot)$ and $\Omega(\cdot)$ notation to suppress the dependence on α in the current overview.

the maximum clique-size. Now, we use *bidimensionality* arguments to show that, either the graph has $\Omega(\sqrt{k}) \times \Omega(\sqrt{k})$ grid as a minor, or the treewidth of the graph is bounded by $\mathcal{O}_\alpha(\sqrt{k})$. In the former case, we can conclude that it is a no-instance, whereas in the latter case we can perform dynamic programming on the tree decomposition of width $\mathcal{O}_\alpha(\sqrt{k})$ to solve the problem. This leads to the following results.

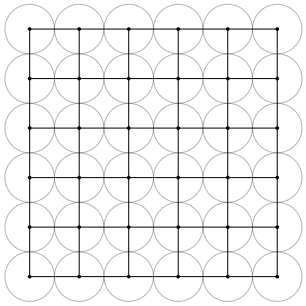
▶ **Theorem 1.** *There exists an algorithm that solves an instance* $\mathcal{I} = (P, k, \alpha)$ *of* k-Shrinking to Independence *in time* $2^{\mathcal{O}((\frac{1}{\alpha})^2\sqrt{k})} \cdot n^{\mathcal{O}(1)}$.

▶ **Theorem 2.** *There exists an algorithm that solves an instance* $\mathcal{I} = (P, k, \alpha)$ *of* k-Shrinking to Acyclicity *in time* $(\frac{1}{\alpha}k)^{\mathcal{O}((\frac{1}{\alpha})^2\sqrt{k})} \cdot n^{\mathcal{O}(1)}$.

Connectivity. Now we turn to the subexponential FPT algorithm for k-Shrinking to Connectivity, which is one of the main results of the paper. This algorithm also begins with the elimination of large cliques. However, already for this step, our arguments are substantially more involved compared to all previously known subexponential-time algorithms for problems on geometric intersection graphs that also eliminate large cliques (see, e.g., [31, 29, 28, 30, 40]). Specifically, for all of these problems, the existence of a large clique trivially implies that we have a yes-instance (or a no-instance) at hand. However, for k-Shrinking to Connectivity, the proof is more complicated. Further, and perhaps more importantly, even once we have proved that every optimal solution contains "most" vertices of a clique, and we have "guessed" (by branching) which these vertices are, we can neither shrink their corresponding disks (since then, we might be left with a clique just as large as it was before) nor delete them. In particular, deletion is incorrect since it might turn a no-instance into a yes-instance, and vice versa (see Figure 4a). Fortunately, we are able to prove that it is possible to delete all of the guessed set except for a carefully chosen subset of a few representatives from it.



(a) There is a solution that shrinks some of the disks in the clique; however, shrinking *all* the disks in the clique results in the disconnected graph (due to the red disks).



(b) Each disk corresponding to the vertices of the grid minor is "touching" its neighbors, and shrinking any of the disks results in a disconnected graph.

Figure 4 Illustrative figures for k-Shrinking to Connectivity.

The next step of our proof handles cycles (or, more precisely, closed walks) that we term *non-empty*. Specifically, we denote some disks as unshrinkable; at the beginning, these are simply the disks that cannot be shrunk without violating connectivity. Then, for a cycle, we consider the "area that it encloses", and identify two sets of disks of interest: the disks intersecting this area (the *relevant set*), and the disks strictly in the interior of this area (the *interior set*). If all the disks in the relevant set are unshrinkable, then the cycle is termed

unshrinkable, and if the interior set is non-empty, then the cycle is termed non-empty. Then, we prove that, given an unshrinkable cycle, it is safe to delete its entire interior set (which, in case of a non-empty cycle, yields progress). Additionally, if the interior set of a cycle is non-empty and contains a shrinkable disk, then the cycle is called *reducible*; we note that a shrinkable non-empty cycle might *not* be reducible. We prove that if we have a large enough collection of reducible cycles whose relevant sets are disjoint, then we have a yes-instance at hand.

Thirdly, we turn to deal with the case where we have a large grid as minor – here, it will also become clear why we had to analyze non-empty cycles. Again, let us remark that all previously known subexponential-time algorithms for problems on geometric intersection graphs, when encountering the case of a large grid as a minor, can simply say Yes or No. For us, having a large grid minor does not imply that we have a yes-instance (or a no-instance), although it may seem so at first glance – indeed, we might even be dealing with an "extreme" case where all of the disks that compose the grid minor are unshrinkable (see Figure 4b). To overcome this difficulty, first of all, we do not consider just any grid minor, but a grid minor such that the closed walks corresponding (according to the minor model) to its 4-cycles have some particular embedding on the plane. Then, we are able to show that within such a grid minor, one can identify either an unshrinkable non-empty closed walk (and, then, apply a reduction rule) or a large enough collection of reducible cycles whose relevant sets are disjoint (and, then, output Yes and terminate). Putting everything together, our algorithm works as follows. As long as we have a large clique, we branch to eliminate most of it. Then, based on a known result, we know that either we have a large grid minor of a particular form (and, then, proceed as described in the previous paragraph) or that the treewidth of the graph is small. In this latter case, we can simply use dynamic programming to directly solve the problem. Thus, we obtain the following result.

▶ **Theorem 3.** *There exists an algorithm that solves an instance $\mathcal{I} = (P, k, \alpha)$ of* k-Shrinking to Connectivity *in time* $(\frac{k}{\alpha})^{\mathcal{O}((\frac{1}{\alpha})^2 \cdot k^{3/4})} \cdot n^{\mathcal{O}(1)}$.

Expanding to achieve Connectivity. Surprisingly, it turns out that, unlike its shrinking counterpart, a generalization of k-Expanding to Connectivity is W[1]-hard. In this generalization, we may only expand the disks centered at a specified subset of points; whereas the disks centered at the rest of the points are fixed to be unit disks. We show this using a reduction from Covering Points by Unit Disks, which is known to be W[1]-hard [43]. This result shows that not all natural problems of interest in this geometric graph modification model are fixed-parameter tractable. [4]

2.2 Polynomial Kernels and Compressions and their Consequences

Now we discuss our results regarding polynomial kernels and compressions for k-Shrinking to Independence/Acyclicity and their minimization variants. We also discuss how these results also enable us to design FPT algorithms for the minimization variants, using a combination of partial enumeration of the solution space of the kernels, and linear

[4] We remark that our algorithmic results (FPT algorithms, kernels, etc.) for the other problems – namely, k-Shrinking to Independence/Acyclicity/Connectivityand their variants – also hold for such a generalization, where a subset of disks cannot be shrunk. In fact, our algorithm for k-Shrinking to Independence supposes the presence of unshrinkable disks, as there, specifically, it simplifies the write-up. Thus, the preceding remark remains valid. However, for the sake of simplicity of presentation of the other results, we decided to stick with the special cases where any disks can be shrunk.

programming. Note that the subexponential FPT algorithms for k-Shrinking to Independence/Acyclicity discussed in the previous section work directly on the original instances and do not rely on the kernels.

2.2.1 (Min) k-Shrinking to Independence

We show the following result.

▶ **Theorem 4.** [Informal] *There exists a partial kernel for k-Shrinking to Independence (resp. for Min k-Shrinking to Independence), where the number of points in the resulting instance is $\mathcal{O}(k)$.*

Note that in this result, we are only able to bound the number of points in the resulting instance by $\mathcal{O}(k)$. However, the number of bits required to encode the coordinates of each point may still be unbounded. Therefore, we only obtain a *partial kernel* for these two problems.

Since k-Shrinking to Independence is a "discrete" problem, we can in fact extend this argument to obtain a (fully) polynomial kernel. Here, we need to additionally show that the geometric information that is relevant for solving k-Shrinking to Independence can be encoded using polynomially many bits in k. Thus, we obtain the following result.

▶ **Theorem 5.** *There exists a (true) polynomial kernel for k-Shrinking to Independence, paramterized by k.*

FPT Algorithm for Min k-Shrinking to Independence. Note that usually the existence of (partial) kernels is sufficient to immediately conclude that the problem is FPT by enumerating all possible solutions. However, for the case of Min k-Shrinking to Independence, even if one fixes a subset of disks of size k to be shrunk (i.e., a *partial solution*), the new radius of each disk is a real number in the range $[\alpha, 1]$. Thus, the number of possible solutions is infinite. Here, we enumerate all partial solutions, and use *linear programming* [5] to check whether there exists a radius assignment of cost at most μ, resulting in an edgeless graph. Thus, we obtain the following result.

▶ **Theorem 6.** *There exists an algorithm that decides an instance $\mathcal{I} = (P, k, \alpha, \mu)$ of* Min k-Shrinking to Independence *in time $k^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$. Additionally, if $\mathcal{I}$ is a yes-instance, the algorithm can return an* optimal solution *for $\mathcal{I}$.*

2.2.2 (Min) k-Shrinking to Acyclicity

We first give an overview of our result regarding the (partial) polynomial compressibility of Min k-Shrinking to Acyclicity, which, along with the FPT algorithm for the problem (as stated subsequently) is the second main result of the paper.

▶ **Theorem 7** (Informal)**.** *There exists a partial polynomial compression for* Min k-Shrinking to Acyclicity *into an annotated graph problem, where the size of the resulting instance is bounded by a polynomial in k, assuming each real number in the instance can be stored using $\mathcal{O}(1)$ bits.*

[5] Since our LP involves real numbers, we cannot use standard LP solvers that run in (weakly) polynomial time. However, since we work in Real RAM model, we can use an algebraic LP solver such as [18, 44] to find an optimum solution, resulting in the algorithm with the claimed running time bound. We thank a reviewer for pointing this out. More details on this can be found in the full version.

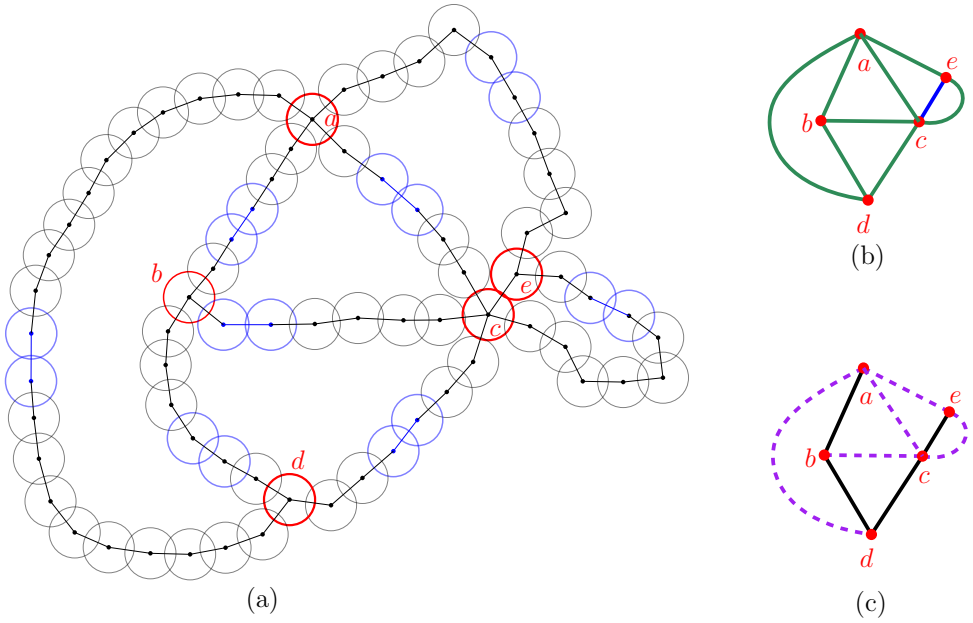


Figure 5 (a) Given intersection graph after removing the vertices of degree at most 1. Vertices of degree at least 3 are shown in red. (b) Resulting multigraph H on vertices of degree at least 3, obtained after replacing degree-2 paths with reducible edges (shown in green) and one of the two edges ce is an original edge (blue). Annotations of green edges include, for example, distance between farthest pairs of disks in the corresponding path (shown in blue in figure (a)). (c) A particular guess for the edges of R (dashed purple), such that $H - R$ (graph induced on black edges) is acyclic. For each edge of R, we guess the number of disks that will be shrunk to remove this edge. Then, an optimal values of the shrunk radii that is compatible with this guess is found using linear programming. For example, such a solution may shrink the disk centered at c (removing bc, ac), e (removing ae and together with c, the edge ce), and the two blue disks in the path corresponding to ad.

We first state relatively straightforward reduction rules. First, we argue that, if any vertex in the initial intersection graph has a "large" degree (i.e., larger than some ck), then it implies the existence of a clique of size at least $k+3$ – note that no solution that is allowed to shrink at most k disks can remove all the edges of such a clique, which implies that we have a no-instance. Thus, either the maximum degree in the given intersection graph is bounded by $\mathcal{O}(k)$, or we can say No. Further, we can safely delete all vertices of degree at most one as well as acyclic components from the graph. So far, these arguments bear resemblance to those used to obtain a polynomial kernel for Feedback Vertex Set. From this point onward, however, we deviate significantly from the standard arguments, and need to rely on the inherent geometric nature of the problem.

We now handle each connected component of the graph separately. If a connected component is a simple cycle, then observe that an optimal solution shrinks either one or two disks, depending on the distances between the consecutive vertices in the cycle. We can compute the optimal choice and get rid of all such isolated cycles. Next, we aim to handle "long" degree-2 paths in a component. Note that an optimal solution shrinks $\ell \leq 2$ disks from such a path. We replace such a path in the intersection graph with a single edge, called a *reducible edge*, that is annotated with an optimal choice for shrinking $0 \leq \ell \leq 2$ disks from the original path. The remaining edges in the graph are *original* edges, and we annotate each edge with the Euclidean distance between its two endpoints.

Thus, we obtain a graph H (in fact, in the actual construction it will be a multigraph). Further, the minimum degree of H is 3, and the maximum degree of H is $\mathcal{O}(k)$. At this point, we can argue that, if the given instance is a yes-instance, then $|V(H)| = \mathcal{O}(k^2)$ and $|E(H)| = \mathcal{O}(k^3)$. The task of determining whether the original instance is a yes-instance of Min k-Shrinking to Acyclicity is equivalent to determining whether H is a yes-instance of an appropriately defined graph problem, say Π. Note that although the size of H is bounded by $\mathcal{O}(k^3)$, the edges of H are annotated with relevant distances denoting optimal choices, and thus can be real numbers in general. Thus, we can only infer a (partial) polynomial compression from Min k-Shrinking to Acyclicity into Π.

We can use the compressed instance of Π to design an FPT algorithm for Min k-Shrinking to Acyclicity, as follows. Recall that $|V(H)| = \mathcal{O}(k^2)$ and $|E(H)| = \mathcal{O}(k^3)$. Thus, any acyclic subgraph of H with vertex set $V(H)$ must contain at most $|V(H)| - 1 = \mathcal{O}(k^2)$ edges. We iterate over all such edge sets $F \subseteq E(H)$ inducing an acyclic subgraph – note that there are at most $k^{\mathcal{O}(k^2)}$ guesses. This implies that each edge in $R = E(H) \setminus F$ must be removed by the solution; note that $|R| = \mathcal{O}(k^3)$. Further, an original edge of R can be removed by shrinking either one or both of its endpoints (3 choices), whereas a reducible edge can be removed by shrinking 1 or 2 disks in the corresponding path, and these choices are encoded in the annotation. Thus, there are a constant number of choices for removal of each edge of R, which results in $2^{\mathcal{O}(k^3)}$ further guesses. For each set of guesses, we can determine an optimal choice for each such guess using the annotated information, and formulate a linear program (LP) that captures the best choices. By solving the LP, in time $2^{\mathcal{O}(k^3 \log k)}$, we can find a minimum-cost solution that shrinks at most k disks that is compatible with all the shrinking choices, if one exists. Finally, we return the best solution found over all $2^{\mathcal{O}(k^3)}$ guesses, which leads to the following result.

▶ **Theorem 8.** *There exists an algorithm that takes as an input of* Min k-Shrinking to Acyclicity*, and in time* $2^{\mathcal{O}(k^3 \log k)} \cdot n^{\mathcal{O}(1)}$ *time, either correctly concludes that it is a no-instance; or finds a solution of the smallest cost that shrinks at most k disks.*

The aforementioned approach also yields a polynomial compression for k-Shrinking to Acyclicity into an intermediate multigraph problem. However, since k-Shrinking to Acyclicity is a "discrete" problem, one can, in fact, encode all the relevant geometric information using polynomially bits in k, resulting in a polynomial kernel for the problem.

▶ **Theorem 9.** *There exists a polynomial kernel for* k-Shrinking to Acyclicity *parameterized by the number of disks k that one is allowed to shrink.*

2.3 Approximation Schemes for (Min) k-Shrinking to Independence

Finally, we design EPTASes, i.e., efficient polynomial-time approximation schemes for (Min) k-Shrinking to Independence. For k-Shrinking to Independence, for a fixed $\varepsilon > 0$, our algorithm either correctly concludes that the given instance is a no-instance, or returns a solution that shrinks at most $(1 + \varepsilon)k$ disks to radius α. For Min k-Shrinking to Independence, our algorithm returns a *bicriteria* EPTAS. That is, it either correctly concludes that the given instance is a no-instance, or returns a solution of cost at most $(1+\varepsilon)\mu$ that shrinks at most $(1+\varepsilon)k$ disks to a radius in $[\alpha, 1]$. Thus, we obtain the following theorem.

▶ **Theorem 10.** *There exists a bicriteria EPTAS (resp. EPTAS) for* Min k-Shrinking to Independence *(resp.* k-Shrinking to Independence*) running in time* $2^{\mathcal{O}(\frac{1}{\varepsilon}\log(\frac{1}{\varepsilon}))} \cdot n^{\mathcal{O}(1)}$.

These results are obtained using the well-known *shifting technique* [6, 34] to reduce the problem to bounded-size instances, at the expense of a small loss in the approximation factor. Each bounded-size instance is then solved optimally. We give an overview for our bicriteria EPTAS for Min k-Shrinking to Independence where the arguments are more involved; those for k-Shrinking to Independence are relatively simpler. First, we subdivide the instance into sub-instances induced by $\mathcal{O}(1/\varepsilon) \times \mathcal{O}(1/\varepsilon)$ squares, such that the number of optimal disks intersecting the boundaries of the squares is at most εk, and the total sum of changes in the radii of such disks is at most $\varepsilon\mu$. In order to solve each subproblem, we further partition the square into smaller grid cells, such that each cell induces a clique in the given UDG. If such a clique contains at most c/ε disks, then we guess the exact subset of disks that are shrunk by an optimal solution. Otherwise, if the clique contains more than c/ε disks, then we mark all disks as *potentially shrinkable*. Note that all except at most one disk from each clique must be shrunk by any solution. Thus, we can argue that the total number of disks that are shrunk in the latter case is at most εk. Now, for each guess of the *potentially shrinkable disks*, we use linear programming to find an optimal solution that is only allowed to shrink the potentially shrinkable disks, in order to result in an edgeless graph. Finally, we need to combine the solutions of subproblems in a careful manner – since the disks near the boundaries of the cell appear in multiple subproblems. We achieve this using dynamic programming over the optimal solutions computed for the bounded-size instances.

As mentioned earlier, we defer the formal descriptions of algorithmic and hardness results to the full version. We proceed directly to the conclusion and directions for future research.

3 Conclusion and Future Research

In this paper, we initiated the study of graph modification by *scaling of disks* from the perspective of Parameterized Complexity. Specifically, we focused on modifying the connectivity properties of the graph while shrinking or expanding a subset of disks. The (Min) k-Shrinking to Independence/Acyclicity problems correspond to shrinking at most k disks to achieve a completely disconnected (i.e., edgeless), and minimally connected (i.e., acyclic) intersection graph, respectively. Whereas k-Shrinking to Connectivity (resp. k-Expanding to Connectivity) corresponds to shrinking at least (resp. at most) k disks while retaining (resp. achieving) connectivity in the intersection graph.

Besides our NP/W[1]-hardness results for the above problems, we also gave several algorithmic results. For (Min) k-Shrinking to Independence, we designed (partial) polynomial kernels, FPT algorithms, and (bicriteria) approximation schemes. For Min k-Shrinking to Independence, we use a novel combination of linear programming with ideas from parameterized complexity and geometric approximation respectively. The FPT algorithm is obtained by first using kernelization to derive a bounded-size instance, and then using LP to solve this instance; whereas the approximation algorithm uses a combination of the classical shifting technique along with LP. We also obtained similar results for (Min)k-Shrinking to Acyclicity; however our arguments and techniques are more sophisticated due to "non-local" nature of the problem. Finally, for k-Shrinking to Connectivity, we introduced a novel decomposition theorem that returns a grid minor of the disk graph along with a particular kind of embedding on the plane, and used this theorem to design a subexponential FPT algorithm for the problem. Similar ideas also gave subexponential FPT algorithms for k-Shrinking to Independence/Acyclicity.

Analogous to the results obtained for k-Shrinking to Independence, it is natural to ask whether k-Shrinking to Connectivity admits (partial) polynomial kernel, and approximation schemes. Furthermore, it is also possible to define cost-optimization version of k-Shrinking to Connectivity and study it from the perspective of fixed-parameter and approximation algorithms.

At a broader level, we believe that a significant portion of our contribution lies in the conceptual realm, with the goal of pioneering a new domain within the field of geometric graph modification problems. Furthermore, it opens up a multitude of potential directions for future research. In this context, we propose a few specific avenues to explore.

- **Modification to other graph Classes:** In addition to graph modification to achieve edgelessness or connectivity, it is intriguing to consider modifying the given geometric graph to belong to other families of graphs, such as acyclic graphs (for minimal connectivity assurance), cluster graphs (for clustering purposes), cliques (for direct communication facilitation), and bounded-degree graphs (which generalize edgeless graphs).
- **Various geometric intersection graphs as input:** Our study primarily focused on the simplest geometric intersection graphs, specifically disk graphs, where graph modification problems often become tractable within the realm of Parameterized Complexity despite being NP-hard. Beyond disk graphs, various other geometric intersection graphs, such as string graphs and map graphs, present substantial differences. Additionally, we can extend our investigations to three-dimensional spaces (e.g., unit ball graphs and more general ball graphs). Therefore, a natural research direction is to explore geometric modifications for graph classes within these broader and distinct geometric intersection graph categories.
- **New geometric modification operations:** While we utilized scaling as a quintessential example of a geometric modification operation, other geometric operations, like movement (recently studied in [25, 26]) or rotation, may be more suitable depending on the context and the nature of the geometric objects under consideration. Furthermore, one might contemplate simultaneously shifting, scaling, and rotating objects (or applying any combination of two out of these three operations). This parallels research on graph modification problems where significant efforts have been made to concurrently address edge deletions and insertions. Beyond these combinations, defining and researching other geometric operations can be highly relevant based on the objectives and the class of geometric intersection graphs at hand. This includes operations like smoothing and dimension reduction.
- **Meta-theorems.** While the study of research questions on a problem-to-problem basis can be sometimes challenging, yet interesting in its own right, the ultimate goal is often to prove meta-theorems that simultaneously assert or refute, say, membership in FPT or the existence of a polynomial kernel, for a host of problems. For example, we may claim, for a whole class of problems, that if a specific constraint X in the definition of any problem in the class can be expressed in some logic (e.g., Monadic Second Order Logic), then the problem is FPT, or that if and only if the input graph (or one of the input graphs) has bounded treewidth, then the problem is FPT, or that if the problem admits an OR-cross-composition from an NP-hard problem, then it has no polynomial kernel unless the polynomial hierarchy collapses. We refer to the books [23, 20, 32] for examples of such results. Clearly, after gaining a better understanding of the problem class introduced in this paper, the pursuit of such meta-theorems represents an intriguing avenue for further research.

References

1 Ankush Acharyya, Minati De, Subhas C Nandy, and Bodhayan Roy. Range assignment of base-stations maximizing coverage area without interference. *Theoretical Computer Science*, 804:81–97, 2020. doi:10.1016/J.TCS.2019.10.044.

2 Akanksha Agrawal, Daniel Lokshtanov, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Feedback vertex set inspired kernel for chordal vertex deletion. *ACM Trans. Algorithms*, 15(1):11:1–11:28, 2019. doi:10.1145/3284356.

3 Akanksha Agrawal, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Interval vertex deletion admits a polynomial kernel. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1711–1730, 2019. doi:10.1137/1.9781611975482.103.

4 Jochen Alber and Jiří Fiala. Geometric separation and exact solutions for the parameterized independent set problem on disk graphs. *Journal of Algorithms*, 52(2):134–151, 2004. doi:10.1016/J.JALGOR.2003.10.001.

5 Greg Aloupis, Mirela Damian, Robin Y. Flatland, Matias Korman, Özgür Özkan, David Rappaport, and Stefanie Wuhrer. Establishing strong connectivity using optimal radius half-disk antennas. *Comput. Geom.*, 46(3):328–339, 2013. doi:10.1016/J.COMGEO.2012.09.008.

6 Brenda S. Baker. Approximation algorithms for np-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994. doi:10.1145/174644.174650.

7 Ahmad Biniaz, Prosenjit Bose, Paz Carmi, Anil Maheshwari, J Ian Munro, and Michiel Smid. Faster algorithms for some optimization problems on collinear points. *arXiv preprint*, 2018. arXiv:1802.09505.

8 Ivan Bliznets, Marek Cygan, Pawel Komosa, Lukáš Mach, and Michał Pilipczuk. Lower bounds for the parameterized complexity of minimum fill-in and other completion problems. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1132–1151. SIAM, 2016.

9 Ivan Bliznets, Fedor V Fomin, Marcin Pilipczuk, and Michał Pilipczuk. Subexponential parameterized algorithm for interval completion. *ACM Transactions on Algorithms (TALG)*, 14(3):35, 2018.

10 Édouard Bonnet and Pawel Rzazewski. Optimality program in segment and string graphs. *Algorithmica*, 81(7):3047–3073, 2019. doi:10.1007/s00453-019-00568-7.

11 Marin Bougeret, Bart M. P. Jansen, and Ignasi Sau. Bridge-depth characterizes which structural parameterizations of vertex cover admit a polynomial kernel. In *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, pages 16:1–16:19, 2020. doi:10.4230/LIPICS.ICALP.2020.16.

12 Gruia Calinescu and Peng-Jun Wan. Range assignment for high connectivity in wireless ad hoc networks. In *International Conference on Ad-Hoc Networks and Wireless*, pages 235–246. Springer, 2003. doi:10.1007/978-3-540-39611-6_21.

13 Yixin Cao. Linear recognition of almost interval graphs. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1096–1115. SIAM, 2016. doi:10.1137/1.9781611974331.CH77.

14 Yixin Cao and RB Sandeep. Minimum fill-in: Inapproximability and almost tight lower bounds. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 875–880. SIAM, 2017.

15 Ioannis Caragiannis, Christos Kaklamanis, and Panagiotis Kanellopoulos. Energy-efficient wireless network design. *Theory of Computing Systems*, 39(5):593–617, 2006. doi:10.1007/S00224-005-1204-8.

16 Paz Carmi and Matthew J. Katz. Power assignment in radio networks with two power levels. *Algorithmica*, 47(2):183–201, 2007. doi:10.1007/S00453-006-1230-1.

17 Erin W. Chambers, Sándor P. Fekete, Hella-Franziska Hoffmann, Dimitri Marinakis, Joseph S. B. Mitchell, Srinivasan Venkatesh, Ulrike Stege, and Sue Whitesides. Connecting a set of circles with minimum sum of radii. *Comput. Geom.*, 68:62–76, 2018. doi:10.1016/j.comgeo.2017.06.002.

18 Timothy M. Chan. Improved deterministic algorithms for linear programming in low dimensions. *ACM Trans. Algorithms*, 14(3):30:1–30:10, 2018. doi:10.1145/3155312.

19 Christophe Crespelle, Pål Grønås Drange, Fedor V. Fomin, and Petr A. Golovach. A survey of parameterized algorithms and the complexity of edge modification. *Comput. Sci. Rev.*, 48:100556, 2023. doi:10.1016/j.cosrev.2023.100556.

20 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

21 Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. A framework for eth-tight algorithms and lower bounds in geometric intersection graphs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 574–586, 2018. doi:10.1145/3188745.3188854.

22 Mark de Berg, Sándor Kisfaludi-Bak, and Gerhard J. Woeginger. The complexity of dominating set in geometric intersection graphs. *Theor. Comput. Sci.*, 769:18–31, 2019. doi:10.1016/J.TCS.2018.10.007.

23 Rodney G Downey and Michael R Fellows. *Fundamentals of parameterized complexity*, volume 4. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

24 David Eppstein. Maximizing the sum of radii of disjoint balls or disks. *arXiv preprint*, 2016. arXiv:1607.02184.

25 Fedor V. Fomin, Petr A. Golovach, Tanmay Inamdar, Saket Saurabh, and Meirav Zehavi. Kernelization for spreading points. In Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman, editors, *31st Annual European Symposium on Algorithms, ESA 2023, September 4-6, 2023, Amsterdam, The Netherlands*, volume 274 of *LIPIcs*, pages 48:1–48:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.ESA.2023.48.

26 Fedor V. Fomin, Petr A. Golovach, Tanmay Inamdar, and Meirav Zehavi. (re)packing equal disks into rectangle. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 60:1–60:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.60.

27 Fedor V. Fomin, Tien-Nam Le, Daniel Lokshtanov, Saket Saurabh, Stéphan Thomassé, and Meirav Zehavi. Subquadratic kernels for implicit 3-hitting set and 3-set packing problems. *ACM Trans. Algorithms*, 15(1):13:1–13:44, 2019. doi:10.1145/3293466.

28 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Decomposition of map graphs with applications. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, pages 60:1–60:15, 2019. doi:10.4230/LIPICS.ICALP.2019.60.

29 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Finding, hitting and packing cycles in subexponential time on unit disk graphs. *Discret. Comput. Geom.*, 62(4):879–911, 2019. doi:10.1007/S00454-018-00054-X.

30 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Eth-tight algorithms for long path and cycle on unit disk graphs. In *36th International Symposium on Computational Geometry, SoCG 2020, June 23-26, 2020, Zürich, Switzerland*, pages 44:1–44:18, 2020. doi:10.4230/LIPICS.SOCG.2020.44.

31 Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Bidimensionality and geometric graphs. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1563–1575, 2012. doi:10.1137/1.9781611973099.124.

32 Fedor V Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: theory of parameterized preprocessing*. Cambridge University Press, 2019.

33 Panos Giannopoulos, Christian Knauer, and Sue Whitesides. Parameterized complexity of geometric problems. *The Computer Journal*, 51(3):372–384, 2008. doi:10.1093/COMJNL/BXM053.

34 Dorit S. Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32(1):130–136, 1985. doi:10.1145/2455.214106.

35 Bart Jansen. Polynomial kernels for hard problems on disk graphs. In *Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 310–321. Springer, 2010. doi:10.1007/978-3-642-13731-0_30.

36 Bart MP Jansen and Marcin Pilipczuk. Approximation and kernelization for chordal vertex deletion. *SIAM Journal on Discrete Mathematics*, 32(3):2258–2301, 2018. doi:10.1137/17M112035X.

37 Ariel Kulik and Hadas Shachnai. Analysis of two-variable recurrence relations with application to parameterized approximations. *CoRR*, abs/1911.02653, Accepted to Symposium on Foundations of Computer Sciences (FOCS 2020), 2019. arXiv:1911.02653,.

38 Jason Li and Jesper Nederlof. Detecting feedback vertex sets of size k in $O^*(2.7^k)$ time. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 971–989, 2020. doi:10.1137/1.9781611975994.58.

39 Errol L Lloyd, Rui Liu, Madhav V Marathe, Ram Ramanathan, and Sekharipuram S Ravi. Algorithmic aspects of topology control problems for ad hoc networks. *Mobile Networks and applications*, 10(1):19–34, 2005. doi:10.1023/B:MONE.0000048543.95178.F5.

40 Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, Jie Xue, and Meirav Zehavi. Subexponential parameterized algorithms on disk graphs (extended abstract). In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 2005–2031, 2022. doi:10.1137/1.9781611977073.80.

41 Dániel Marx. Efficient approximation schemes for geometric problems? In *Algorithms - ESA 2005, 13th Annual European Symposium, Palma de Mallorca, Spain, October 3-6, 2005, Proceedings*, pages 448–459, 2005. doi:10.1007/11561071_41.

42 Dániel Marx. Parameterized complexity of independence and domination on geometric graphs. In *Parameterized and Exact Computation, Second International Workshop, IWPEC 2006, Zürich, Switzerland, September 13-15, 2006, Proceedings*, pages 154–165, 2006. doi:10.1007/11847250_14.

43 Dániel Marx. On the optimality of planar and geometric approximation schemes. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 338–348. IEEE Computer Society, 2007. doi:10.1109/FOCS.2007.50.

44 Jirí Matousek, Micha Sharir, and Emo Welzl. A subexponential bound for linear programming. *Algorithmica*, 16(4/5):498–516, 1996. doi:10.1007/BF01940877.

45 Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Contraction decomposition in unit disk graphs and algorithmic applications in parameterized complexity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1035–1054, 2019. doi:10.1137/1.9781611975482.64.

46 Paolo Santi. Topology control in wireless ad hoc and sensor networks. *ACM computing surveys (CSUR)*, 37(2):164–194, 2005. doi:10.1145/1089733.1089736.

47 Stéphan Thomassé. A $4k^2$ kernel for feedback vertex set. *ACM Trans. Algorithms*, 6(2):32:1–32:8, 2010. doi:10.1145/1721837.1721848.

48 P-J Wan, Gruia Călinescu, X-Y Li, and Ophir Frieder. Minimum-energy broadcasting in static ad hoc wireless networks. *Wireless Networks*, 8(6):607–617, 2002. doi:10.1023/A:1020381720601.

Combinatorial Pen Testing (Or Consumer Surplus of Deferred-Acceptance Auctions)

Aadityan Ganesh
Princeton University, NJ, USA

Jason Hartline
Northwestern University, Evanston, IL, USA

Abstract

Pen testing is the problem of selecting high-capacity resources when the only way to measure the capacity of a resource expends its capacity. We have a set of n pens with unknown amounts of ink and our goal is to select a feasible subset of pens maximizing the total ink in them. We are allowed to learn about the ink levels by writing with them, but this uses up ink that was previously in the pens.

We identify optimal and near optimal pen testing algorithms by drawing analogues to auction theoretic frameworks of deferred-acceptance auctions and virtual values. Our framework allows the conversion of any near optimal deferred-acceptance mechanism into a near optimal pen testing algorithm. Moreover, these algorithms guarantee an additional overhead of at most $(1 + o(1)) \ln n$ in the approximation factor to the omniscient algorithm that has access to the ink levels in the pens. We use this framework to give pen testing algorithms for various combinatorial constraints like matroid, knapsack, and general downward-closed constraints, and also for online environments.

2012 ACM Subject Classification Theory of computation → Theory and algorithms for application domains; Theory of computation → Approximation algorithms analysis; Theory of computation → Algorithm design techniques

Keywords and phrases Pen testing, consumer surplus, money-burning, deferred-acceptance auctions

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.52

Related Version *Full Version*: `https://arxiv.org/abs/2301.12462`

Funding *Jason Hartline*: This work is supported in part by the NSF award CCF-2229162.

1 Introduction

Pen testing is the problem of selecting high capacity resources when the only way to measure the capacity of the resource expends its capacity [23]. We show that any ascending-price auction can be converted into an equivalent pen testing algorithm. We apply the auction theoretic framework of virtual values to identify optimal pen testing algorithms. This connection allows many existing results from auction theory to be applied to pen testing and gives optimal and near optimal pen testing algorithms in combinatorial and online environments.

The pen testing problem of Qiao and Valiant [23] is the following. We have a set of pens with varying amounts of remaining ink and we want to choose a pen with the largest amount of ink left. We have access to the distribution of ink levels in these pens, but we are only allowed to learn about the ink levels by writing with the pens to test them. While writing gives information about whether there is still ink left, it uses up ink that was previously in the pen. Pens that are expended due to testing can be discarded without incurring any penalty. The combinatorial pen testing problem generalizes the pen testing problem to selecting a feasible subset of pens, e.g., according to a matroid or knapsack constraint, to maximize the total remaining ink in the chosen subset.

© Aadityan Ganesh and Jason Hartline;
licensed under Creative Commons License CC-BY 4.0

Deferred-acceptance auctions [21] formalize the notion of ascending-price mechanisms for allocating goods to strategic agents. These are mechanisms that greedily reject the least promising agent by increasing the price for getting allocated at each stage. For instance, to auction off one good, the deferred-acceptance mechanism keeps increasing the price, rejecting agents whenever their value for the good falls below the price until exactly one agent remains in the auction. The *surplus* of the mechanism is the sum of the values of all the allocated agents and the *consumer surplus* is the surplus minus the payments made by the agents.

We provide a black box reduction to convert any surplus optimal (or near optimal) deferred-acceptance auction into an optimal (or near optimal) pen testing algorithm with equivalent performance guarantees. In this reduction, pens and their ink levels are analogous to agents and their values. Payments correspond to the ink used up via testing, and as a consequence, the consumer surplus of the deferred-acceptance auction corresponds to the residual ink in the pen testing environment. Therefore, optimizing residual ink reduces to designing consumer-surplus-optimal deferred-acceptance mechanisms.

Qiao and Valiant [23] propose comparing the performance of a given pen testing algorithm against the *omniscient benchmark*, the performance of an omniscient algorithm that has access to the ink levels in the pens. The omniscient benchmark can thereby choose the optimal feasible subset of pens without having to burn any ink (i.e, payments in the auction environment), and is thus, analogous to the surplus optimal (sealed-bid) mechanism in the corresponding auction environment. Thus, the *omniscient approximation*, the ratio between the omniscient benchmark to the residual ink in the pen testing algorithm, is analogous to the ratio between the optimal surplus and the consumer surplus of the corresponding deferred-acceptance auction.

In the literature, near optimal deferred-acceptance mechanisms are evaluated against optimal (sealed bid) mechanisms, an upper bound to the optimal deferred-acceptance mechanism. Define the *standard benchmark* as the optimal (sealed-bid) mechanism for consumer surplus and let the *standard approximation* $\gamma(n)$ be the ratio between the standard benchmark and the consumer surplus of a given deferred-acceptance mechanism. Note that the standard benchmark does not have a natural pen testing analog, but is a much tighter upper bound on the optimal pen testing algorithm than the omniscient benchmark. Thus, the reduction to deferred-acceptance mechanism design enables a much tighter approximation bound for the optimal pen testing algorithm.

The reduction framework decomposes into two steps for designing near optimal deferred-acceptance mechanisms. One of the two steps involves proving upper bounds on the standard approximation $\gamma(n)$. In the other step, for any given feasibility constraint, we consider the *optimal omniscient approximation*, the ratio between the omniscient benchmark (here, the optimal surplus) and the standard benchmark (here, the optimal consumer surplus). As we will argue in Theorem 1, the omniscient approximation $\pi(n)$ is at most the product $\gamma(n)\,\zeta(n)$ between the two approximation ratios. At a high level, $\gamma(n)$ captures the loss in surplus due to running a deferred-acceptance auction and $\zeta(n)$ captures the loss in surplus due to burning all payments. Beyond consumer surplus optimization, the reduction framework applies more generally to optimization problems with two natural benchmarks, a weaker standard benchmark and much more powerful omniscient benchmark.

▶ **Theorem 1.** *Consider a combinatorial pen testing environment with n pens, the analogous auction environment with an optimal omniscient approximation $\zeta(n)$, and a deferred-acceptance mechanism with a standard approximation equal to $\gamma(n)$. Then, there is a pen testing algorithm that is a $\gamma(n)$-approximation to the standard benchmark and $\pi(n) = \gamma(n)\,\zeta(n)$-approximation to the omniscient benchmark.*

In the first step of our framework, we bound the optimal omniscient approximation $\zeta(n)$ under various feasibility constraints. Generally, remember that the optimal omniscient approximation $\zeta(n)$ is the ratio between the omniscient benchmark and the standard benchmark, and for our specific application, it corresponds to the loss in surplus due to payments getting burnt instead of getting transferred to the auctioneer. We show that $\zeta(n) \leq (1 + o(1)) \ln n$ in any combinatorial environment. Our result tightens the best known upper bound on $\zeta(n)$ for most settings of interest. Fotakis et al. [13] show that the optimal omniscient approximation is at most $\frac{2}{\ln 2}(1 + o(1)) \ln m$, where m is the number of feasible outcomes. Our bound improves their bound by removing the $\frac{2}{\ln 2}$ constant and by replacing m by n, which is usually smaller.[1] Hartline and Roughgarden [18] show that $\zeta(n) \leq \frac{2}{\ln 2}(1 + o(1)) \ln \frac{n}{k}$ when any k out of the n agents can be allocated the good simultaneously, i.e, any k pens can be chosen in the corresponding pen testing environment. We improve their bound by a multiplicative factor of at least $1.36(1 - o(1))$. We also prove that $\zeta(n) \geq (0.577 + \ln \frac{n}{k})$ for the k-identical goods setting, which we show is tight up to $(1 + o(1))$ terms when $k = o(n)$.

Our upper bound for $\zeta(n)$ is via a reduction from multi-agent environments to a single-agent environment, similar in flavour to Feng et al. [12]. In a single-agent environment where the agent is allocated with an ex-ante allocation probability q, we show that the ratio of the optimal surplus to the optimal consumer surplus is at most $1 - \ln q$. Our reduction allows us to convert the $(1 - \ln q)$-approximation in the single-agent setting to a $(1 + o(1) \ln n$-approximation for any multi-agent environment.

We reduce designing multi-agent approximation mechanisms to single-agent approximations with ex-ante constraints q bounded away from 0. The approach is inspired by Feng et al. [12], who prove a similar reduction when the single-agent approximation is bounded above by a constant for all $q \in [0, 1]$. Their approach cannot be directly applied to the optimal omniscient approximation of consumer surplus because the single-agent approximation $(1 - \ln q)$ approaches ∞ as $q \to 0$. We adapt ideas from Hartline and Taggart [17] to prove a reduction to the single-agent setting with an ex-ante constraint $q \in [1/n, 1]$. In this range, the single-agent approximation ratio is at most $(1 + \ln n)$. The resulting construction establishes an optimal omniscient approximation of $(1 + o(1)) \ln n$ for multi-agent environments with arbitrary feasibility constraints.

The second step of the reduction framework from pen testing to deferred-acceptance auctions upper bounds the standard approximation $\gamma(n)$, the ratio between the optimal consumer surplus of a sealed-bid mechanism and the consumer surplus of a given deferred-acceptance auction. Through Myerson's virtual value framework [22], we argue that the above is equivalent to finding an upper bound for the optimal expected surplus divided by the expected surplus of some deferred-acceptance auction. Surplus optimal deferred-acceptance auctions are known for k-identical goods and more general matroid feasibility constraints [21, 4]. We give a 2-approximate deferred-acceptance mechanism for knapsack constraints. Beyond matroids and knapsacks, Feldman et al. [11] give a logarithmic approximate deferred-acceptance auction for any downward-closed feasibility constraint. We observe an inherent downward-closure in the pen testing problem that allows the auction of Feldman et al. [11] to be extended to all combinatorial constraints. The set of pens chosen by any pen testing algorithm can be padded by adding (possibly expended) pens to always output a maximal feasible set.

[1] In some non-downward-closed settings like public projects, $m < n$. For example, $m = 2$ for building a bridge that serves everyone. Either the bridge is built or not. However, $m \geq n$ in all downward-closed environments. Indeed, for each bidder, we assume there exists an outcome where they are included; otherwise, the bidder can be dropped from consideration altogether.

Table 1 Prior bounds (if any, references indexed by superscripts) and our upper bounds. $n =$ number of agents, $m =$ number of feasible outcomes. Note that $m \geq n$ for all the environments listed in the table. The bounds for optimal omniscient approximation ζ and omniscient approximation π are normalized by a $(1 + o(1))$ factor. Prior work: [*]Hartline and Roughgarden [18]; [**] Milgrom and Segal [21], Bikhchandani et al. [4]; [§]Fotakis et al. [13]; [†] Feldman et al. [11]; [††] Samuel-Cahn [25], Chawla et al. [6]; [§§]Kleinberg and Weinberg [19]; [‡] Qiao and Valiant [23]; [‡‡] Chawla et al. [6], Yan [26]. All bounds in the table also hold for the underlying auction environment. Pen testing algorithms with the same guarantees as downward-closed constraints can be achieved for arbitrary combinatorial constraints.

Environment	$\frac{\zeta(n)}{(1+o(1))}$	$\gamma(n)$	$\frac{\pi(n)}{(1+o(1))}$
Select $k = o(n)$	$\frac{2}{\ln 2} \ln \frac{n}{k}$ [*] $\ln \frac{n}{k}$	1 [**]	$\ln \frac{n}{k}$
Select $k = \Theta(n)$	$\frac{2}{\ln 2} \ln \frac{n}{k}$ [*] $2.27\,(0.577 + \ln \frac{n}{k})$	1 [**]	$2.27\,(0.577 + \ln \frac{n}{k})$
Matroids	$\frac{2}{\ln 2} \ln m$ [§] $\ln n$	1 [**]	$\ln n$
Knapsack	$\frac{2}{\ln 2} \ln m$ [§] $\ln n$	2	$2 \cdot \ln n$
General Downward-Closed	$\frac{2}{\ln 2} \ln m$ [§] $\ln n$	$O(\log \log m)$ [†]	$O(\log n \log \log m)$
Select 1 Online Oblivious	$\ln n$	2 [††]	$O(\log n)$ [‡] $2 \cdot \ln n$
Select 1 Online Sequential	$\ln n$	$\frac{e}{e-1}$ [‡‡]	$O(\log n)$ [‡] $\frac{e}{e-1} \cdot \ln n$
Select 1 Online IID	$\ln n$	$\frac{e}{e-1}$ [‡‡]	$e \ln n$ [‡] $\ln n$
Matroid Online Oblivious	$\ln n$	2 [§§]	$2 \cdot \ln n$
Matroid Online Sequential	$\ln n$	$\frac{e}{e-1}$ [‡‡]	$\frac{e}{e-1} \cdot \ln n$

Online posted-price mechanisms are a special case of deferred-acceptance auctions. Online environments are ones where an irrevocable decision regarding choosing a pen needs to be taken before testing the next one. We consider the settings where the order of testing pens is adversarially chosen (the oblivious version, *cf.* prophet inequalities, [6]) and where the algorithm can choose the order (the sequential version, *cf.* correlation gap, [26]). With our analysis of the optimal omniscient approximation ζ and known bounds on the standard approximation γ of online pricing mechanisms, the reduction framework yields better performance guarantees than the $O(\log n)$ bounds by Qiao and Valiant [23] by constant factors for these environments.

The results detailed above are summarized in Table 1.

Our reduction from pen testing to deferred-acceptance auction design improves bounds from previous literature for all the environments that we discuss; however, it does not always yield a tight bound. For the online IID environment (the online environment where the ink levels of all pens are drawn IID), we identify an algorithm with an omniscient approximation $\pi(n)$ that is a constant factor less than $\gamma(n)\,\zeta(n)$.

Our results and techniques are of interest beyond pen testing. Our upper bound of $(1 + o(1)) \ln n$ on the optimal omniscient approximation $\zeta(n)$ is the ratio between the optimal surplus and the optimal consumer surplus, and is a mechanism design problem of independent interest [18, 13, 15]. Recall that the surplus created in an auction equals the sum of the consumer surplus and the revenue generated by the auctioneer. However, when payments made by the winners are burnt instead of being transferred to the auctioneer, the total surplus created is just the consumer surplus. The need to burn payments can be diverse. For instance:

1. Making monetary transactions is not desirable or repugnant, like in setting prices for welfare schemes for vulnerable populations. Ordeal-based mechanisms are employed in such scenarios, where agents make non-transferable non-monetary payments, like spending time waiting in a line to avail the provisions provided in the welfare scheme [20, 3, 5].
2. In computer systems like data packet routing networks, where technological infeasibility to collect payments necessitates charging non-monetary payments [10, 9, 18].

3. The auctioneer cannot be trusted to truthfully implement the auction, like in a blockchain, where a miner runs an auction to decide which set of transaction goes into the next block it proposes. To stop the miner from colluding with agents or from benefiting from injecting fake bids, it becomes essential to burn a substantial portion of the transaction fees collected from the agents [24, 7, 14].

The technique to obtain the $(1 - \ln q)$-approximation in the single-agent environment can have further applications. We express consumer surplus optimization as a linear program and explicitly compute the worst-case distribution. This provides a proof technique which automatically gives tight approximation guarantees. For example, we reuse the technique to compute the worst-case distribution for the omniscient approximation for the online IID environment. We improve the bound obtained from the reduction framework to a multiplicative approximation ratio within an additive 1 of the H_n lower bound of Qiao and Valiant [23] (where H_n is the n^{th} harmonic number).

Paper Organization. We make the reduction from the pen-testing problem to designing deferred-acceptance auctions formal in Section 2. We execute the first step of the reduction framework – upper bounding the optimal omniscient approximation $\zeta(n)$ – in Section 3. We bound the approximation ratio between the surplus and consumer surplus in a single-agent environment in Section 3.1 and discuss the reduction from multi-agent to single-agent settings in Section 3.2. The second step of the reduction from pen testing algorithms to deferred-acceptance mechanisms is discussed in Section 4, where we review deferred-acceptance mechanisms for various environments and discuss the corresponding bounds we get for pen testing. Finally, we revisit the online IID single item environment and achieve a tight (up to $o(1)$ terms) bound in Section 5.

2 The Pen Testing Problem and Ascending-Price Auctions

In this section, we formally define the combinatorial pen testing problem, generalized from Qiao and Valiant [23], and explore its connections to ascending-price mechanisms.

▶ **Definition 2.** *A pen testing instance is described by a set $N = \{1, 2, \ldots, n\}$ of pens with unknown ink levels $v_1, \ldots, v_n$, each v_i drawn independently from the distribution F_i, and a subset $\mathcal{P}$ of the powerset of the pens denoting feasible subsets of pens. The residual ink level u_i of pen i is initiated to v_i. We can test the pens before making a decision. Test (i, θ_i) is done by writing with pen i for time θ_i. We receive a binary signal at the end of the test:*
1. *If $u_i \geq \theta_i$: The test succeeds. The pen now has an (unobservable) ink level $u_i \leftarrow u_i - \theta_i$.*
2. *If $u_i < \theta_i$: The test fails. The pen now has no ink left, i.e, $u_i \leftarrow 0$.*

We can choose to run multiple tests on the pens. We need to output some feasible subset $P \in \mathcal{P}$ of pens, maximizing

$$\sum\nolimits_{i \in P} u_i.$$

The goal is to optimize the total amount of ink remaining in the set of chosen pens.

Compare a pen testing instance with n pens and a feasibility constraint $\mathcal{P}$ with an n-agent auction under the same feasibility constraint. The original levels of ink correspond to the value each agent gets upon being allocated. The ink spent through testing is analogous to the prices in the auction. Thus, maximizing total residual ink is equivalent to optimizing consumer surplus, i.e, the sum of the values of the winning agents minus the sum of all payments. Note that auctions generally give the auctioneer the added advantage of the

ability to solicit bids from agents. In pen testing algorithms, we only get to learn whether the ink left in the pen is more than some threshold and in doing so, we irrevocably expend ink up to the threshold. This is similar to ascending-price auctions, where the auctioneer irreversibly increases the price faced by each agent, and in doing so, only learns whether the agent has a value at least the price.

Milgrom and Segal [21] describe the wide class of ascending-price auctions called deferred-acceptance mechanisms.

▶ **Definition 3** (Milgrom and Segal [21]; Deferred-Acceptance Auctions). *A deferred-acceptance auction is held across multiple stages. For each stage t, the auction maintains a set of active bidders A_t satisfying $A_1 \supseteq A_2 \supseteq \cdots \supseteq A_t$, where the initial active set A_1 is the set of all bidders. The auction is characterized by a (possibly randomized) pricing rule $\vec{p}$, mapping the history of the auction at stage t to (discriminatory) prices for each agent satisfying $p_i(H_t) \geq p_i(H_{t-1})$ for all agents i and for all histories H_t at stage t arising out of stage $t-1$ history H_{t-1}, i.e, the prices are monotonously non-decreasing for all agents over time. Agents can opt to drop out once the prices are updated in stage t. A_{t+1} is the set of agents in A_t that did not drop out in stage t. The mechanism terminates at stage t when A_t becomes feasible. The agents in A_t are charged according to $\vec{p}(H_t)$.*

The definition from Milgrom and Segal [21] restricts deferred-acceptance mechanisms to be deterministic, which we relax for the purpose of designing pen testing algorithms. From the discussion above, note that any deferred-acceptance mechanism can be converted into a pen testing algorithm, by setting thresholds equal to the prices recommended by the mechanism. Thus, the optimal (near optimal) pen testing algorithm corresponds to the consumer surplus optimal (near optimal) deferred-acceptance auction.

We compare the performance of our deferred-acceptance mechanisms (and the corresponding pen testing algorithms) to the following benchmarks:

- The *omniscient benchmark* [23]: The omniscient benchmark corresponds to the omniscient pen testing algorithm that exactly knows the ink levels in the pens, and thus, does not have to waste ink by writing with them. In the reduction to the auction environment, the omniscient benchmark corresponds to a mechanism that neither loses out on the payments nor has to stick to an increasing trajectory of prices. Thus, the omniscient benchmark corresponds to the surplus optimal sealed-bid mechanism. The *omniscient approximation* $\pi(n)$ corresponds to the ratio between the omniscient benchmark and the performance of a given pen testing algorithm in the pen testing setting, and the omniscient benchmark and the consumer surplus of a given deferred-acceptance auction in the auction environment.
- The *standard benchmark*: The standard benchmark corresponds to the consumer-surplus-optimal sealed-bid mechanism. We define the *standard approximation* $\gamma(n)$ to be the ratio between the standard benchmark and the consumer surplus of a candidate deferred-acceptance mechanism. While the standard benchmark does not have a natural pen testing analogy, observe that the standard benchmark is a tighter upper bound to the optimal pen testing algorithm than the omniscient benchmark. Thus, the standard approximation gives a better upper bound on the ratio between the optimal pen testing algorithm and the residual ink gathered by a given pen testing algorithm.

Now that we have established the consumer surplus is a quantity of interest for pen testing, we review consumer surplus optimization through virtual values in the next section.

2.1 Consumer Surplus Optimization and Virtual Valuations

Let us begin in a single-agent environment with one good. Let the agent's value be drawn from the distribution F. The quantile q of an agent with value $v \sim F$ is the measure with respect to F of stronger values, i.e, $q = 1 - F(v)$. For $q \in [0, 1]$, let $v(q)$ be the inverse demand function of F satisfying $F(v(q)) = 1 - q$. In other words, $Pr_{\hat{v}\sim F}(\hat{v} > v(q)) = q$. Throughout the paper, we assume $v(1) = 0$ and $\int_0^0 v(t)dt = 0$. It can be shown that these assumptions can be made without loss of generality.

Let $V(q) = \int_0^q v(t)dt$ be the price-posting surplus curve. Notice that $V(q)$ is the expected surplus from posting a price $v(q)$ to the agent, thereby allocating the good with probability q. By our assumption on F, $V(0) = 0$. Let $U(q) = V(q) - qv(q)$ be the price-posting consumer surplus curve. Similar to the price-posting surplus curve, $U(q)$ is the consumer-surplus from posting a price $v(q)$. Note that $0 \leq U(0) \leq V(0) = 0$, and thus, $U(0) = 0$.

Let $u(q) = U'(q) = -qv'(q)$ be the marginal price-posting consumer surplus curve. Note that v is monotonously non-increasing, and hence its derivative v' is non-positive. Thus, $u(q) \geq 0$ and U is monotone non-decreasing. A quick note on derivatives: whenever $v'(q)$ is well defined, $u(q)$ is well defined. At other points, $u(q)$ can be calculated using any sub-derivative of v. At points q of discontinuity of v, $v'(q) = -\infty$ and thus $u(q) = \infty$.

▶ **Theorem 4** (Myerson [22]). *In a Bayesian incentive-compatible mechanism with allocation rule y and payment rule p (over quantiles), the expected consumer surplus of an agent satisfies*

$$E_{q\sim U[0,1]}[v(q)y(q) - p(q)] = E_{q\sim U[0,1]}[u(q)y(q)] = E_{q\sim U[0,1]}[-U(q)\,y'(q)] + U(0)y(0)$$

Thus, the expected consumer surplus equals the expected virtual surplus with marginal consumer surplus as virtual values. Myerson [22] states that an allocation rule can be implemented as a truthful auction if and only if the allocation rule is monotone non-increasing in quantile space. Thus, the consumer surplus optimal mechanism optimizes for virtual surplus subject to the allocation rule being monotonously non-increasing.

When the marginal price-posting consumer surplus curve is non-increasing (and hence U is concave), optimizing for virtual surplus automatically ensures monotonicity of the allocation rule. We will call these distributions with non-increasing marginal price-posting consumer surplus curves *consumer surplus regular*.

However, u is not monotone non-increasing for common distributions like the uniform and normal distribution. In fact, u is monotone non-decreasing for these distributions. In such a case, the allocation that pointwise optimizes for virtual surplus might not satisfy monotonicity. Myerson [22] prescribes an ironing procedure to optimize for virtual surplus subject to monotonicity.

1. Construct the concave hull $\overline{U}$ of the price-posting consumer surplus curve U
2. Define $\overline{u}(q) = \overline{U}'(q)$
3. Find the virtual surplus optimal mechanism using $\overline{u}$ as virtual surplus

Since $\overline{U}$ is concave, $\overline{u}$ is non-increasing, and hence optimizing for the ironed virtual surplus is easier. Throughout the paper, we will follow the convention of attaching a bar on top of a curve to describe the ironed curve.

In a multi-agent environment, the interim allocation rule for an agent is the single-agent allocation rule that arises in expectation over the quantiles of all other agents. This captures the perspective of the agent after knowing its value, but before learning the values of the other agents.

▶ **Theorem 5.** *Let y be the interim allocation rule for some agent, satisfying $\frac{d}{dq}y(q) = 0$ for all q such that $U(q) \neq \overline{U}(q)$. Then*

$$E_q[u(q)y(q)] = E_q[\overline{u}(q)y(q)]$$

In other words, if the allocation rule does not differentiate between agents in an ironed region, then the ironed virtual surplus can be used to compute the consumer surplus instead of the actual virtual surplus.

▶ **Theorem 6** (Alaei et al. [1, 2]). *Let y be some allocation rule with interim allocation rule y_i monotonously non-increasing for each agent i. Then, there exists a mechanism $\overline{y}$ with interim allocation rule $\overline{y}_i$ for agent i such that the expected consumer surplus for agent i equals*

$$E_{q\sim U[0,1]}[u_i(q)\,\overline{y}_i(q)] = E_{q\sim U[0,1]}[\overline{u}_i(q)\,y_i(q)]$$

In other words, the expected consumer surplus of $\overline{y}$ is the consumer surplus of the original mechanism if the virtual values were given by $\overline{u}_i$ for agent i instead of u_i

2.2 Near-Optimal Deferred-Acceptance Mechanisms for Consumer Surplus

In this section, we prescribe a recipe similar to Feng et al. [12] to convert any approximately optimal deferred-acceptance mechanism for surplus into an approximately optimal deferred-acceptance mechanism for consumer surplus.

▶ **Definition 7** (Virtual-Pricing Transformation of a Deferred-Acceptance Mechanism). *Let DA^V be a deferred-acceptance mechanism designed to optimize surplus. Let v_i be the inverse demand function and u_i be the virtual value function (for consumer surplus) for agent i. The virtual-pricing transformation on DA^V, denoted by DA^U, implements DA^V in ironed virtual price space, i.e, whenever DA^V posts a price $\hat{v}_i$ to agent i, DA^U posts a price $v_i(\theta_i)$ satisfying*

$$\theta_i = \sup\{\theta : \overline{u}_i(\theta) \geq \hat{v}_i\}$$

We will make some preliminary observations about the transformation:

1. Consider two runs of the transformed mechanism DA^U with some agent having two different values, both corresponding to the same ironed virtual value (which can happen if both these values are within the same ironed interval). From the definition of the transform, the agent faces the smallest price needed to differentiate itself from virtual values smaller than the threshold set by the mechanism. Consequently, the mechanism does not post prices from the middle of an ironed interval. Thus, in both these runs of the mechanism, the agent behaves identically. Since the mechanism does not discriminate between values with the same ironed virtual value, the consumer surplus of the mechanism equals the ironed virtual surplus.
2. If the original mechanism DA^V is a γ-approximation to the optimal surplus, in expectation over all product distributions, then the transformed mechanism DA^U is a γ-approximation to the optimal ironed virtual surplus, and hence, to the optimal consumer surplus.
3. Finally, it is straightforward to see that the transformed mechanism DA^U posts non-decreasing prices to each agent. Hence, DA^U is also a deferred-acceptance mechanism.

Recall that the standard approximation $\gamma(n)$ is the ratio between the consumer surplus of the optimal deferred-acceptance mechanism and our proposed mechanism while the optimal omniscient approximation $\zeta(n)$ is the ratio between the optimal surplus and the optimal consumer surplus. The proof of Theorem 1 is now straightforward.

Proof of Theorem 1. Let OPT^U denote the expected optimal consumer surplus, OPT^V denote the expected optimal surplus, and DA^U denote the expected consumer surplus of the deferred acceptance mechanism in Definition 7. Then,

$$\mathrm{DA}^U \geq \frac{\mathrm{OPT}^U}{\gamma(n)} \geq \frac{\mathrm{OPT}^V}{\gamma(n)\,\zeta(n)}$$

where the first inequality follows from the definition of standard approximation $\gamma(n)$ and bullet 2 in the discussion above, while the second follows from the definition of optimal omniscient approximation $\zeta(n)$. The equivalent pen testing algorithm yields the necessary approximation ratios against the two benchmarks. ◀

In the next section, we analyze the optimal omniscient approximation $\zeta(n)$ for general combinatorial environments.

3 Consumer Surplus versus Surplus

Our approximation procedure follows two steps. In Section 3.1, we show that the optimal surplus is at most $1-\ln q$ times the consumer surplus in a single-agent environment where the ex-ante probability of allocation is constrained to be at most q. In Section 3.2, we move from the single-agent environment to multi-agent environments combining approaches from Feng et al. [12] and Hartline and Taggart [17] to show a $(1+o(1))\ln n$ approximation between the consumer surplus and optimal surplus in n-agent environments with general feasibility constraints.

3.1 The Single-Agent Problem

Recall that V, U and $\overline{U}$ are the price-posting surplus curve, price-posting consumer surplus curve and the ironed consumer surplus curve respectively. In this section, we establish a "closeness property" like in Feng et al. [12].

▶ **Theorem 8.** *For an ex-ante allocation probability $q \in [0,1]$, the ratio of the optimal surplus $V(q)$ to the optimal consumer surplus $\overline{U}(q)$ is at most $1-\ln q$.*

We will begin by proving Theorem 8 for consumer surplus regular distributions, and then extend the result to all distributions.

▶ **Lemma 9.** *For any consumer surplus regular distribution and an ex-ante allocation probability $q \in [0,1]$, the ratio of the optimal surplus $V(q)$ to the optimal consumer surplus $U(q)$ is at most $1-\ln q$.*

Proof. Conditional on generating a surplus $V(q)$, we want to compute the distribution that minimizes consumer surplus $U(q)$.

$$\begin{aligned} U(q) &= \int_0^q u(t)dt. \\ V(q) &= U(q) + qv(q) \\ &= \int_0^q u(t)dt - q\int_q^1 v'(t)dt \\ &= \int_0^q u(t)dt + q\int_q^1 \frac{u(t)}{t}dt. \end{aligned} \tag{1}$$

We substitute $u(t) = -tv'(t)$ in the last equation.

Simultaneously, we enforce $u(t) \geq 0$, and $u(t)$ is non-increasing (since the distribution is consumer surplus regular), $v(1) = 0$ and $V(0) = 0$.

Finding the minimum consumer surplus distribution reduces to the following program.

$$\min \int_0^q u(t)dt$$

subject to

$$\int_0^q u(t)dt + q\int_q^1 \frac{u(t)}{t}dt = V(q)$$

$u(t) \geq 0, u(t)$ is monotonously non-increasing

$v(1) = 0, V(0) = 0$

The constraints in the last line can be enforced after solving for u.

Consumer surplus regularity dictates $u(t) \geq u(q)$ for $t \leq q$. Hence, minimizing $\int_0^q u(t)dt$ would correspond to setting $u(t) = u(q)$ for $t \leq q$. Similarly, $u(t) \leq u(q)$ for $t \geq q$. Minimizing $\int_0^q u(t)dt$ would mean minimizing $V(q) - q\int_q^1 \frac{u(t)}{t}dt$ which is achieved by setting $u(t) = u(q)$ for $t \geq q$. Let $u(0) = u(t) = u(1) = u$. For a constant marginal price-posting consumer surplus curve,

$$U(q) = \int_0^q u(t)dt = q\,u$$

$$V(q) = \int_0^q u(t)dt + q\int_q^1 \frac{u(t)}{t}dt = [q - q\ln q]\,u$$

Thus, for any consumer surplus regular distribution, $\frac{V(q)}{U(q)} \leq [1 - \ln q]$. The constant marginal consumer surplus curve that achieves this ratio corresponds to the exponential distribution. ◀

▶ **Lemma 10.** *Let the consumer surplus curves $U, \hat{U}$ satisfy $\hat{U}(0) = 0$ and $\hat{U}(t) \geq U(t)$ for all $t \in [0, 1]$. Then, for the corresponding surplus curves, $\hat{V}(q) \geq V(q)$.*

Proof. We rewrite equation (1) as follow.

$$\begin{aligned} V(q) &= \int_0^q u(t)dt + q\int_q^1 \frac{u(t)}{t}dt \\ &= U(q) + q\Big[\frac{U(t)}{t}\Big]_{t=q}^{t=1} + q\int_q^1 \frac{U(t)}{t^2}dt \\ &= qU(1) + q\int_q^1 \frac{U(t)}{t^2}dt \end{aligned} \tag{2}$$

The second equality is obtained through integrating by parts. Increasing U pointwise clearly results in an increase in V. ◀

Proof of Theorem 8. The optimal consumer surplus for an ex-ante allocation probability q equals $\overline{U}(q)$.

Consider the distribution with price-posting consumer surplus curve $\overline{U}$. Let $\overline{V}$ be the price-posting surplus curve corresponding to $\overline{U}$. From Lemma 10 and Lemma 9,

$$\frac{V(q)}{\overline{U}(q)} \leq \frac{\overline{V}(q)}{\overline{U}(q)} \leq 1 - \ln q$$

The second inequality holds from Lemma 9, since $\overline{V}$ is the price-posting surplus curve of a consumer surplus regular distribution (we made $\overline{U}$ concave through ironing). ◀

See the full online version for a brief discussion on the surplus generated by the single-agent optimal consumer surplus auction.

3.2 Multi-Agent Environments

In this section, we convert the single-agent bound from Theorem 8 into an upper bound on the ratio between the optimal surplus and optimal consumer surplus (aka the optimal omniscient approximation) for multi-agent environments. We outline our approach before delving into the proof. Consider the interim allocation rule y_i for agent i in the surplus optimal mechanism OPT^V. Without loss of generality, assume $y_i(1) = 0$. The expected surplus for agent i equals

$$E_{q\in U[0,1]}[y_i(q)\, v_i(q)] = \left[y_i(q)\, V_i(q)\right]_{q=0}^{q=1} + E_{q\in U[0,1]}[-y_i'(q)\, V(q)] = E_{q\in U[0,1]}[-y_i'(q)\, V(q)]$$

This equality comes from integrating by parts ($y_i(q)V_i(q)$ vanishes at both, $q = 0$ and 1, since $V_i(0) = 0$ and $y_i(1) = 0$). Suppose there exists a constant α such that $V_i(q) \le \alpha\, \overline{U}_i(q)$ for all $q \in [0, 1]$, then,

$$E_{q\in U[0,1]}[-y_i'(q)\, V(q)] \le E_{q\in U[0,1]}[-y_i'(q)\, \alpha\, \overline{U}(q)] = \alpha \cdot E_{q\in U[0,1]}[y_i(q)\, \overline{u}_i(q)]$$

The above inequality connects the expected surplus to the expected ironed consumer surplus of the interim allocation rule y_i. Since $\frac{d}{dq} y_i$ need not be equal to 0 whenever $U(q) \neq \overline{U}(q)$, $E_{q\in U[0,1]}[y_i(q)\, \overline{u}_i(q)] \neq E_{q\in U[0,1]}[y_i(q)\, u_i(q)]$. But, Theorem 6 shows the existence of a mechanism $\overline{y}$ with expected consumer surplus of each agent i equal to $E_{q\in U[0,1]}[y_i(q)\, \overline{u}_i(q)]$. Thus, the expected consumer surplus of $\overline{y}$ is an α-approximation to the optimal surplus. The consumer surplus optimal mechanism OPT^U will only do better.

Unfortunately, as $q \to 0$, the bound from Theorem 8 of $\frac{V_i(q)}{\overline{U}_i(q)} \le (1 - \ln q)$ diverges to ∞. Thus, we need to show that the loss from ignoring the small (strong) quantiles is not much. Suppose we find a mechanism $\hat{y}$ with interim allocation $\hat{y}_i$ for agent i, near optimal surplus, and $\hat{y}_i'(q) = 0$ for $q \in [0, \epsilon]$. Then, $y_i'(q) \times V(q) = y_i'(q) \times U(q) = 0$ in this region. For $q \in [\epsilon, 1]$, we know $V(q) \le (1 - \ln \epsilon) \times \overline{U}(q)$. Thus,

$$(1 - \ln \epsilon) \times \text{OPT}^U \ge \text{surplus}(\hat{y})$$

Hartline and Taggart [17] give the construction of such a mechanism $\hat{y}$, originally intended to design good mechanisms with only sample access to the distributions of bidders' bids.

▶ **Definition 11** (Hartline and Taggart [17]; ϵ-buffering rule)**.** *Given an allocation rule $\vec{y} = (y_1, y_2, \ldots, y_n)$ and a quantile $\epsilon \in [0, 1]$, the ϵ-buffering rule for $\vec{y}$ simulates y with quantiles transformed on each agent as follows:*

- *Top inflate: for any $q_i \in [0, \epsilon]$, return 0*
- *For any $q_i \in [\epsilon, 1 - \epsilon]$, return $\frac{q_i - \epsilon}{1 - 2\epsilon}$*
- *Bottom deflate: for any $q_i \in [1 - \epsilon, 1]$, return 1*

In essence, the top inflate branch makes the ϵ-buffering rule treat all agents with a small quantile as if they had quantile 0 and thus, $y_i'(q) = 0$ for $q \in [0, \epsilon]$ (the bottom deflate branch is unnecessary for the analysis of monotone payoff curves like surplus V and consumer surplus U; see Remark 15).

▶ **Theorem 12** (Hartline and Taggart [17])**.** *Let $\hat{y}$ be the ϵ-buffering rule of the surplus optimal mechanism OPT^V. Then, $\text{OPT}^V \le \frac{1}{(1 - \frac{\epsilon}{1-\epsilon})\,(1-\epsilon)\,(1-2n\epsilon)} \times \text{surplus}(\hat{y})$.*

▶ **Theorem 13.** *Consider an n-agent environment with an arbitrary feasibility constraint. The optimal omniscient approximation is at most $\frac{1 - \ln \epsilon}{(1 - \frac{\epsilon}{1-\epsilon})\,(1-\epsilon)\,(1-2n\epsilon)}$.*

Proof. The proof follows by combining Theorem 12 with the discussion above. Let $\hat{y}$ be the ϵ-buffering rule of the surplus optimal mechanism OPT^V. Then,

$$\text{OPT}^V \leq \frac{1}{(1-\frac{\epsilon}{1-\epsilon})\,(1-\epsilon)\,(1-2n\epsilon)} \times \text{surplus}(\hat{y}) \leq \frac{1-\ln\epsilon}{(1-\frac{\epsilon}{1-\epsilon})\,(1-\epsilon)\,(1-2n\epsilon)} \times \text{OPT}^U$$

◀

▶ **Corollary 14.** *In n-agent environments with an arbitrary feasibility constraint, the optimal omniscient approximation $\zeta(n) \leq (1+o(1))\ln n$.*

Proof. Setting $\epsilon = \frac{1}{n\ln n}$ in Theorem 13, we get

$$\begin{aligned}
\zeta(n) &\leq \frac{1-\ln\frac{1}{n\ln n}}{(1-\frac{\frac{1}{n\ln n}}{1-\frac{1}{n\ln n}})\,(1-\frac{1}{n\ln n})\,(1-2n\cdot\frac{1}{n\ln n})} \\
&= \frac{1+\ln n+\ln\ln n}{(1-\frac{1}{n\ln n-1})\,(1-\frac{1}{n\ln n})\,(1-\frac{2}{\ln n})} \\
&= \frac{n\ln n-1}{n\ln n-2}\cdot\frac{n\ln n}{n\ln n-1}\cdot\frac{\ln n-2}{\ln n}\cdot\frac{1+\ln n+\ln\ln n}{\ln n}\times\ln n \\
&= (1+o(1))\ln n
\end{aligned}$$

◀

▶ Remark 15. The ϵ-buffering rule of Hartline and Taggart [17] can be implemented without a bottom deflate branch. This would strengthen the approximation guarantee to $\frac{1}{(1-\epsilon)\,(1-n\epsilon)}$. However, substituting $\epsilon = \frac{1}{n\ln n}$ will not result in a bound tighter than $(1+o(1))\ln n$.

▶ Remark 16. Fotakis et al. [13] show that the optimal omniscient approximation as a function of the number of outcomes m in the feasibility constraint is at most $\frac{2}{\ln 2}(1+o(1))\ln m$. Corollary 14 improves their bound whenever the feasibility constraint has cardinality at least $\sqrt{n}$. Note that all downward-closed constraints have at least n feasible outcomes. For such feasibility constraints, Corollary 14 yields an exponential improvement. As the number of feasible outcomes become exponential in n, their bound declines to $O(n)$.

▶ Remark 17. The upper bound from Corollary 14 matches the lower bound of Qiao and Valiant [23]. In the single-item environment with values of agents drawn IID from the exponential distribution, they show that the optimal surplus is $H_n = (1+o(1))\ln n$ times the optimal consumer surplus, where H_n is the n^{th} harmonic number.

3.3 The k-Identical Goods Environment

Theorem 13 and Corollary 14 give an over-arching bound for the optimal omniscient approximation $\zeta(n)$ that holds in any environment. However, specific environments admit better approximations. For instance, we will give a tighter bound for $\zeta(n)$ for the k-identical goods environment, where any set of at most k-agents can be allocated simultaneously.

▶ **Theorem 18.** *In the n-agent k-identical goods environment, the optimal omniscient approximation satisfies*

1. $\zeta(n) \leq (1+o(1))\ln\frac{n}{k}$ *when* $k = o(n)$, *and*
2. $\zeta(n) \leq 2.27\,(1+o(1))(0.577+\ln\frac{n}{k})$, *when* $k = \Theta(n)$.

To prove Theorem 18, we make use of the following lemma, whose proof we defer to the end of the section.

▶ **Lemma 19.** *Consider the special case with n agents and k identical goods. The optimal omniscient approximation $\zeta(n) \leq \frac{1}{(1-\frac{1}{\sqrt{2\pi k}})}\cdot(1-\ln\epsilon)(1+[\frac{n}{k}-1]\,\epsilon)$ for any $\epsilon > 0$.*

Lemma 19 is particularly useful when $k = \Theta(n)$. For a constant ϵ, $\frac{1}{(1-\frac{1}{\sqrt{2\pi k}})}$ is $(1 + o(1))$ while $(1 - \ln \epsilon)(1 + [\frac{n}{k} - 1]\,\epsilon)$ is a constant.

Proof of Theorem 18. We show the theorem separately for the three cases, when $k = \Theta(1)$, $k = \omega(1)$ but $o(n)$, and $k = \Theta(n)$.

- $k = \Theta(1)$: The proof is a direct consequence of Corollary 14, where we show $\zeta(n) \leq (1 + o(1)) \ln n = (1 + o(1)) \ln \frac{n}{k}$. The equality follows since k is a constant.
- $k = \omega(1)$, but $o(n)$: By setting $\epsilon = \frac{1}{\frac{n}{k}(1+\ln \frac{n}{k})}$ in Lemma 19, we get

$$\zeta(n) \leq \left(\frac{1}{1-\frac{1}{\sqrt{2\pi k}}}\right) \times \left(1 + \frac{1}{1 + \ln \frac{n}{k}}\,[1 - \frac{k}{n}]\right) \times (1 + \ln \tfrac{n}{k} + \ln(1 + \ln \tfrac{n}{k}))$$

  The first term is $(1 + o(1))$ since $k = \omega(1)$. The second and the third terms are $(1 + o(1))$ and $(1 + o(1)) \ln \frac{n}{k}$ respectively, since $k = o(n)$. Summarizing, $\zeta(n) \leq (1 + o(1)) \ln \frac{n}{k}$.

The above two cases conclude the proof of the first part of the theorem. The next case clinches the proof of the second part.

- $k = \Theta(n)$: We set $\epsilon = \frac{1}{\frac{n}{k}(1+\ln \frac{n}{k})}$ as in the previous case. Once again $\left(\frac{1}{1-\frac{1}{\sqrt{2\pi k}}}\right) = (1+o(1))$ since $k = o(1)$. We plot the ratio between our upper bound and the lower bound $(0.577 + \ln \frac{n}{k})$ from Lemma 20 as a function of $\frac{k}{n}$ in Figure 1a to observe that the ratio is at most 2.27, exactly as needed. We skip a more detailed numerical analysis substantiating this observation. ◀

We modify the lower bound from Qiao and Valiant [23] sketched in Remark 17 to get an analogous result for the k-identical goods environment.

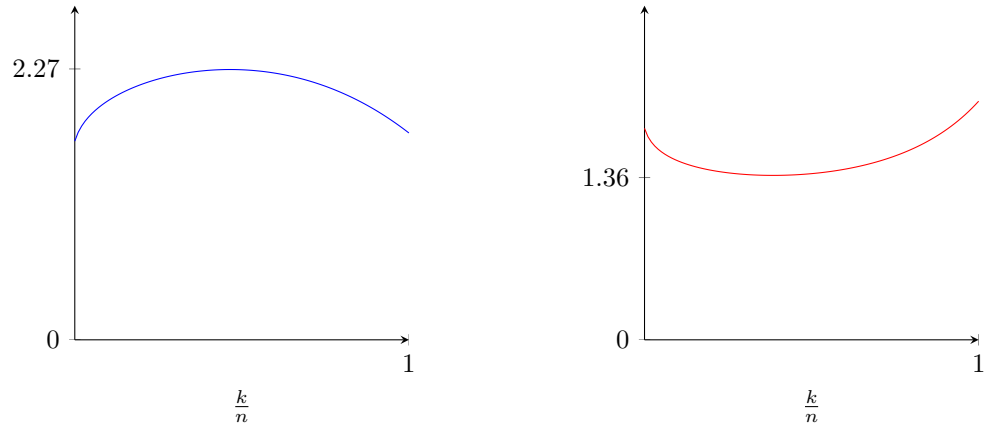
▶ **Lemma 20.** *Consider the n-agent k-identical goods environment with values drawn IID from the exponential distribution with mean 1. Assume, for convenience, that $\frac{n}{k}$ is an integer. The optimal omniscient approximation $\zeta(n) \geq H_{\frac{n}{k}} \geq (0.577 + \ln \frac{n}{k})$.*

Proof. The exponential distribution with mean 1 has a constant marginal consumer surplus curve $u(q) = 1$, and thus, the optimal consumer surplus mechanism allocates to some set of k agents to get an expected consumer surplus equal to k.

Now, consider the following mechanism. Group agents into k batches of $\frac{n}{k}$ agents each, and award the agent with the highest value in each group. Each group is now identical to a single item environment with $\frac{n}{k}$ agents, and thus, generates an expected surplus of $H_{\frac{n}{k}}$ (see Remark 17 on the example sketched in [23]). The k batches together produce an expected surplus equal to $k \cdot H_{\frac{n}{k}}$. The optimal mechanism creates a larger surplus and hence, $\zeta(n)$ is at least $H_{\frac{n}{k}} \geq (0.577 + \ln \frac{n}{k})$. ◀

Thus, the upper bound from Theorem 18 is tight up to $(1 + o(1))$ terms when $k = o(n)$ and up to a multiplicative $2.27\,(1 + o(1))$ when $k = \Theta(n)$. The best known bound for the optimal omniscient approximation is due to Hartline and Roughgarden [18], where they show $\zeta(n) \leq \frac{2}{\ln 2}\,(\ln 2 + \ln \frac{n}{k}) \approx 2.88(\ln 2 + \ln \frac{n}{k})$. Apart from closing the $\frac{2}{\ln 2}$ gap between their upper bound and the lower bound when $k = o(n)$, our bound is an improvement by a factor of at least $1.36(1 - o(1))$ when $k = \Theta(n)$ (Figure 1b).

We conclude the section with the proof of Lemma 19. We will adopt an approach similar to the proof of Theorem 13. Increasing the allocation to the agents with a small quantile so that $y_i(q_i) = y_i(0)$ for $q_i \in [0, \epsilon]$ vastly increases the consumer surplus of i without noticeably decreasing the surplus of the mechanism. We consider online posted-price mechanisms with the constraint $y_i'(q_i) = 0$ whenever $q_i \leq \epsilon$ for all agents i. Posted-price mechanisms are known to achieve a near optimal surplus and we bound the loss in surplus due to the additional constraint.



(a) **(b)**

Figure 1 We compare our upper bound against the lower bound and the best known upper bound when $k = \Theta(n)$. Figure (a) plots the ratio between our upper bound from Theorem 18, normalized by $(1 + o(1))$ and our lower bound from Lemma 20, $(0.577 + \ln \frac{n}{k})$. The maximum value of the curve is less than 2.27, and hence, our upper bound can potentially be improved by a factor of at most $2.27\,(1 + o(1))$. Figure (b) plots the ratio between the current best known bound $\frac{2}{\ln 2}\,(\ln 2 + \ln \frac{n}{k})$ and our bound from Theorem 18, again normalized by $(1 + o(1))$. The minimum value taken by the ratio is greater than 1.36, and hence, our bound improves the best known bound by a factor of at least $1.36(1 - o(1))$.

Consistent with prior literature on online sequential price-posting [6, 26] we will approximate the performance of the mechanism against the optimal ex-ante relaxed mechanism EAR. In the ex-ante relaxation, the feasibility constraint is relaxed to bind ex-ante rather than ex-post. In other words, the expected number of agents served by the mechanism must be at most k. The optimal ex-ante mechanism obtains a larger surplus than OPT^V, since OPT^V is also a feasible ex-ante mechanism. Note that, if agent i is being allocated with probability q_i by EAR, it would allocate to the q_i smallest quantiles (quantiles corresponding to the largest values) without violating the ex-ante constraint, thereby getting an expected surplus $V_i(q_i)$ from agent i.

Let Δ be the set of all ex-ante feasible allocations $\vec{q} = (q_1, \ldots, q_n)$ for selling up to k items such that $q_1, \ldots, q_n \in [0, 1]$ and $\sum_{i=1}^{n} q_i \leq k$. Let $\text{EAR}(\vec{q})$ be the ex-ante relaxed mechanism that allocates to the q_i smallest quantiles of agent i. Note that $\text{EAR}(\vec{q})$ is well-defined even if $\vec{q} \notin \Delta$ as long as $q_1, \ldots, q_n \in [0, 1]$. We will abuse notation to denote the surplus from $\text{EAR}(\vec{q})$ by $\text{EAR}(\vec{q})$. Thus, $\text{EAR} = \max_{\vec{q} \in \Delta} \text{EAR}(\vec{q})$.

Yan [26] showed that the following sequential posted-price mechanism $\text{SP}(\vec{q})$ has a surplus comparable to $\text{EAR}(\vec{q})$ for all $\vec{q} \in \Delta$.

1. Order agents in decreasing order of $\frac{V_i(q_i)}{q_i}$, the expected surplus conditioned on selling to agent i.
2. In the order above, offer agent i a price $v_i(q_i)$ until all n agents interact with the mechanism or until all k goods are sold.

As usual, we will abuse notation to denote the surplus of $\text{SP}(\vec{q})$ by $\text{SP}(\vec{q})$.

▶ **Theorem 21** (Yan [26]). *For* $\vec{q} \in \Delta$, $\frac{1}{(1 - \frac{1}{\sqrt{2\pi k}})} \times \text{SP}(\vec{q}) \geq \text{EAR}(\vec{q})$.

We propose a sequential-posted-price-mechanism with near optimal surplus and interim allocation rules satisfying the additional properties outlined above. For a vector $\vec{q} \in \Delta$, let the ϵ-inflated sequential posted-price mechanism $\text{SP}(\vec{q}, \epsilon)$ denote $\text{SP}(\vec{p})$, where $p_i = \max\{q_i, \epsilon\}$.

Observe that if $\mathrm{SP}(\vec{q},\epsilon)$ has an unsold good while interacting with agent i, it always sells the good if the agent has a quantile $q_i \le \epsilon$. Thus, the interim allocation rule y_i is a constant function in $[0,\epsilon]$ and $y_i'(q) = 0$ in this range. Next, we argue that $\mathrm{SP}(\vec{q},\epsilon)$ approximates the surplus of $\mathrm{EAR}(\vec{q})$.

▶ **Lemma 22.** *For all* $\vec{q} \in \Delta$, $\mathrm{EAR}(\vec{q}) \le \frac{(1+[\frac{n}{k}-1]\,\epsilon)}{(1-\frac{1}{\sqrt{2\pi k}})}\,\mathrm{SP}(\vec{q},\epsilon)$

Proof. The proof proceeds in two steps. First, for $\vec{p}$ with $p_i = \max\{q_i,\epsilon\}$, we show that $\vec{p}$ is not too far away from the set of ex-ante feasible allocations Δ. We then show that the projection $\vec{t}$ of $\vec{p}$ back onto Δ satisfies $\mathrm{EAR}(\vec{q}) \le \mathrm{EAR}(\vec{p}) \le \frac{k+(n-k)\,\epsilon}{k} \times \mathrm{EAR}(\vec{t})$ and $\mathrm{SP}(\vec{t}) \le \mathrm{SP}(\vec{p}) = \mathrm{SP}(\vec{q},\epsilon)$.

Step 1. $\sum_{i=1}^n p_i \le k + (n-k)\,\epsilon$.

Without loss of generality, we assume $\sum_{i=1}^n q_i = k$. Increasing q_i so that their sum equals k would not decrease the value of $\sum_{i=1}^n p_i$. The claim reduces to showing the maximum value of $\sum_{i=1}^n (p_i - q_1)$ is at most $(n-k)\,\epsilon$.

Note that $p_i - q_i \le \epsilon$, equality holding exactly when $q_i = 0$. Thus, whenever $q_i < \epsilon$, decreasing them to zero increases the value of $\sum_{i=1}^n (p_i - q_i)$. However, this would contradict $\sum_{i=1}^n q_i = k$. For $q_i \ge \epsilon$, $p_i - q_i = 0$. Increasing q_i to 1 whenever $q_i \ge \epsilon$ does not change the value of $\sum_{i=1}^n (p_i - q_i)$, but allows other values q_j to be set to zero without violating $\sum_{i=1}^n q_i = k$. A simple greedy argument would suggest that $\sum_{i=1}^n (p_i - q_i)$ is maximized when $q_i = 1$ for $1 \le i \le k$ and $q_i = 0$ when $i > k$, in which case, $\sum_{i=1}^n (p_i - q_i) < (n-k)\,\epsilon$.

Step 2. Order (and re-index) agents in decreasing order of $V_i(p_i)$. Consider the ex-ante allocation $\frac{k}{k+(n-k)\,\epsilon} \cdot \mathrm{EAR}(\vec{p})$ that offers agent i with $\frac{k}{k+(n-k)\,\epsilon}$ fraction of a good whenever $q_i \le p_i$. This has a surplus $\frac{k}{k+(n-k)\,\epsilon} \cdot \mathrm{EAR}(\vec{p}) = \sum_{i=1}^n \frac{k}{k+(n-k)\,\epsilon} \cdot V_i(p_i)$ and allocates up to k units, since $\sum_{i=1}^n p_i \le k + (n-k)\,\epsilon$. Let $\vec{t}$ be an alternate ex-ante feasible allocation that allocates $t_i = p_i$ for agents $1,\ldots,m$ for some $m \le n$ until all k items are allocated. $\vec{t}$ redistributes goods from agents with a smaller surplus $V_i(p_i)$ allocated by $\frac{k}{k+(n-k)\,\epsilon} \cdot \mathrm{EAR}(\vec{p})$ to agents with a larger $V_i(p_i)$ to get a surplus

$$\mathrm{EAR}(\vec{t}) = \sum_{i=1}^m V_i(p_i) \ge \sum_{i=1}^n \frac{k}{k+(n-k)\,\epsilon} \cdot V_i(p_i) = \frac{k}{k+(n-k)\,\epsilon} \cdot \mathrm{EAR}(\vec{p})$$

Step 3. $\mathrm{SP}(\vec{t}) \le \mathrm{SP}(\vec{p})$

$\mathrm{SP}\,\vec{t}$ allocates the first m agents with probability p_i and allocates the others with probability 0. $\mathrm{SP}(\vec{p})$ allocates the first m agents with probability p_i and further allocates any unsold good to the others. Thus, $\mathrm{SP}(\vec{t}) \le \mathrm{SP}(\vec{p})$.

Summarizing,

$$\begin{aligned}
\mathrm{EAR}(\vec{q}) &\le \mathrm{EAR}(\vec{p}) \\
&\le \tfrac{k+(n-k)\,\epsilon}{k} \times \mathrm{EAR}(\vec{t}) = (1 + [\frac{n}{k} - 1]\,\epsilon) \times \mathrm{EAR}(\vec{t}) \\
&\le \frac{(1 + [\frac{n}{k} - 1]\,\epsilon)}{(1 - \frac{1}{\sqrt{2\pi k}})} \times \mathrm{SP}(\vec{t}) \\
&\le \frac{(1 + [\frac{n}{k} - 1]\,\epsilon)}{(1 - \frac{1}{\sqrt{2\pi k}})} \times \mathrm{SP}(\vec{p}) = \frac{(1 + [\frac{n}{k} - 1]\,\epsilon)}{(1 - \frac{1}{\sqrt{2\pi k}})} \times \mathrm{SP}(\vec{q},\epsilon)
\end{aligned}$$

The third inequality follows from Theorem 21 ◀

Proof of Lemma 19. Let $\vec{q} \in \Delta$ be such that $\text{EAR} = \text{EAR}(\vec{q})$. From the review on sequential-price-posting and Lemma 22,

$$\text{OPT}^V \leq \text{EAR} = \text{EAR}(\vec{q}) \leq \tfrac{(1+[\frac{n}{k}-1]\,\epsilon)}{(1-\frac{1}{\sqrt{2\pi k}})} \times \text{SP}(\vec{q},\epsilon)$$

Further, $\text{SP}(\vec{q},\epsilon)$ has a constant interim allocation rule for agent i when its quantile q_i is in the range $[0,\epsilon]$. Arguments identical to the proof of Theorem 13 can be used to conclude

$$\zeta(n) = \frac{\text{OPT}^V}{\text{OPT}^U} \leq \tfrac{(1+[\frac{n}{k}-1]\,\epsilon)}{(1-\frac{1}{\sqrt{2\pi k}})} \cdot \frac{\text{SP}(\vec{q},\epsilon)}{\text{OPT}^U} \leq \tfrac{1}{(1-\frac{1}{\sqrt{2\pi k}})} \cdot (1-\ln\epsilon)(1+[\frac{n}{k}-1]\,\epsilon) \qquad \blacktriangleleft$$

4 Pen Testing Corollaries from Deferred-Acceptance Mechanisms

There are many environments in which deferred-acceptance mechanisms are known to be good. By Theorem 1, these imply good pen testing algorithms (up to an additional $\zeta(n)$ factor in the omniscient approximation, i.e, the ratio of the performances of the omniscient algorithm that knows the amount of ink in each pen and our proposed pen testing algorithm). In this section, we discuss a few notable examples. A summary of these results was given previously in Table 1.

Deferred-acceptance mechanisms that achieve the ex-post surplus optimal outcome are known for various feasibility constraints. While the simple auction that uniformly increases the price until exactly k bidders remain active is surplus optimal in the k-identical goods environment, Milgrom and Segal [21] and Bikhchandani et al. [4] generalize the mechanism for matroid feasibility constraints. We get the following corollaries from the above auctions.

▶ **Corollary 23.** *For a pen testing environment with a matroid feasibility constraint, there exists a pen testing algorithm with an omniscient approximation ratio* $(1+o(1))\ln n$.

▶ **Corollary 24.** *When choosing any k of the n pens is feasible, there exists a pen testing algorithm with an omniscient approximation ratio* $(1+o(1))\ln\frac{n}{k}$ *when* $k=o(n)$ *and* $2.27\,(1+o(1))\,(0.577+\ln\frac{n}{k})$ *when* $k=\Theta(n)$.

Dütting et al. [8] initiated the study of prior-free deferred-acceptance approximation mechanisms in environments where deferred-acceptance mechanisms are known to be suboptimal. Their bounds can be improved in settings like ours where prior distributions of the agents' values are known. For general downward-closed constraints with a prior distribution on values, Feldman et al. [11] give a $O(\log\log m)$ approximation to the optimal surplus, where m is the number of maximal feasible sets. Note that $m \leq 2^n$ (every subset of agents might be feasible) and hence, $\log\log m$ is at most $\log n$. Thus, we have a poly-logarithmic approximate pen testing algorithm for any downward-closed feasibility environment.

Note, however, that the pen testing model is inherently downward-closed. Specifically, suppose an algorithm wanted to select a set $P \subseteq \overline{P}$, but P is not feasible while $\overline{P}$ is. Since all pens have non-negative residual ink, even the ones that failed their tests, there is no loss in selecting $\overline{P}$ instead of the set P. Thus, near optimal pen testing algorithms for downward-closed constraints can be extended to give near optimal algorithms for general combinatorial constraints, giving the same performance guarantee as the downward-closed environment.

▶ **Corollary 25.** *For any combinatorial pen testing environment, there exists a pen testing algorithm with an omniscient approximation ratio* $O(\log n\log\log m) = O(\log^2 n)$.

▶ Remark 26. Note that this reduction cannot be used to design deferred-acceptance mechanisms for general feasibility constraints. Allocating to agents that have dropped out of the auction can make the mechanism non-truthful (i.e, staying active till the price reaches the agent's value and dropping out subsequently might not be in the agent's best interests).

The bound for downward-closed and general combinatorial environments can be improved upon in special cases; for example, in the full online version, we give a natural deferred-acceptance mechanism for knapsack constraints that is a 2-approximation. For this knapsack problem, the agents have sizes along with values and feasible subsets are precisely those with a total size at most the knapsack capacity.

▶ **Corollary 27.** *For a pen testing environment with a knapsack feasibility constraint, there exists a pen testing algorithm with an omniscient approximation ratio* $2\,(1 + o(1)) \ln n$.

Online pricing mechanisms are a special case of deferred-acceptance mechanisms; thus, Theorem 1 converts online pricing mechanisms with good surplus into good online pen testing algorithms. For online pen testing problems, every pen can be tested exactly once and must either be selected or discarded immediately after testing and before testing another pen. The algorithm might have the ability to determine the order of testing pens (the sequential version) or the order might be adversarially determined (the oblivious version).

For the sequential pricing problem, Chawla et al. [6] gave an $\frac{e}{e-1}$-approximation mechanism for the single-item environment. Yan [26] connected the sequential pricing problem to the correlation gap and generalized the $\frac{e}{e-1}$-approximation to matroid environments. Both of these papers considered optimizing revenue, but the results can be easily adapted to optimize surplus.

▶ **Corollary 28.** *In the online sequential pen testing problem with a matroid feasibility constraint, there exists a pen testing algorithm that achieves an omniscient approximation ratio* $\frac{e}{e-1} \cdot (1 + o(1)) \ln n$.

Hajiaghayi et al. [16] and Chawla et al. [6] adapt the prophet inequality of Samuel-Cahn [25] to give a non-adaptive 2-approximation for the oblivious price posting problem with 1 good. Kleinberg and Weinberg [19] extend this 2-approximation to matroid environments and any arrival order of agents but with adaptive prices.

▶ **Corollary 29.** *In the online oblivious pen testing problem to select a single pen, there exists a non-adaptive pen testing algorithm that achieves an omniscient approximation ratio* $2(1 + o(1)) \ln n$.

▶ **Corollary 30.** *In the online oblivious pen testing problem with a matroid feasibility constraint, there exists a pen testing algorithm that achieves an omniscient approximation ratio* $2(1 + o(1)) \ln n$.

The bounds of Corollary 28 and Corollary 29 improve on the online pen testing bounds of Qiao and Valiant [23].

5 The Online IID Environment

Consider the special case of the online pen testing problem where the ink levels are drawn independently from the same distribution. We move away from the reduction framework of Theorem 1 to obtain a better bound for this environment.

▶ **Theorem 31.** *In online single-item environments with IID agents, there exists a price-posting strategy with an omniscient approximation at most* $(1 + o(1)) \ln n$*. Equivalently, in the online pen testing problem with IID ink levels, there exists an algorithm that achieves* $\pi(n) \leq (1 + o(1)) \ln n$*.*

First, note that the above theorem is tight. In Remark 17, we saw an instance (originally from [23]) with n IID agents whose values are drawn from the exponential distribution with consumer surplus only an H_n approximation to the optimal surplus. The optimal online consumer surplus cannot do better. Hence, the $(1 + o(1)) \ln n$ approximation ratio in Theorem 31 is tight.

Before proving Theorem 31, we briefly discuss the guarantee obtained through the reduction framework of Theorem 1. For the online environment with IID agents, it is known that $\gamma(n) = \frac{e}{e-1}$ [6]. Thus, the framework would give us an $\frac{e}{e-1} \cdot (1 + o(1)) \ln n$ bound. The direct analysis below gives a tight $H_n + 1 = (1 + o(1)) \ln n$ bound up to an additive factor 1.

Proof of Theorem 31. Similar to the proof of Theorem 8, we will phrase our program as an optimization problem and compute the optimal solution.

The surplus optimal auction always allocates the good to the agent with the highest value (smallest quantile). Let v_{max} be the random variable denoting the value of the winner.

$Pr(v_{\max} \leq v(t)) = (1 - t)^n$

Thus, the expected surplus of the surplus optimal mechanism equals

$$\begin{aligned}
\int_0^1 v(t) \times n(1-t)^{n-1} dt &= \int_0^1 \int_t^1 \frac{u(r)}{r} dr \times n(1-t)^{n-1} dt \\
&= \int_0^1 \frac{u(t)}{t} dt - \int_0^1 \frac{u(t)}{t} (1-t)^n dt \\
&= \int_0^1 \frac{1-(1-t)^n}{t} u(t) dt \\
&= U(1) + \int_0^1 \frac{1-(1-t)^n - nt(1-t)^{n-1}}{t^2} U(t) dt
\end{aligned}$$

The first equality follows by integrating from $u(t) = -tv'(t)$ and $v(1) = 0$. We integrate by parts for the second and fourth equation.

Next, we look at the consumer surplus through price-posting. For a price $v(q)$, we allocate the good if at least one agent has a quantile in $[0, q]$. Conditioned on selling the good, we get an expected consumer surplus of $\frac{U(q)}{q}$. Thus, the expected consumer surplus equals

$$\frac{1-(1-q)^n}{q} \times U(q)$$

As before, we first show Theorem 31 for consumer surplus regular distributions. Without loss of generality, assume that the maximum achievable consumer surplus through anonymous price posting is at most 1. We want to find the consumer surplus regular distribution with the largest surplus.

$$\max U(1) + \int_0^1 \frac{1-(1-t)^n - nt(1-t)^{n-1}}{t^2} U(t) dt$$

subject to

$$\frac{1-(1-q)^n}{q} \times U(q) \leq 1 \text{ for all } q \in [0, 1]$$

U is concave

First, observe that $C(q) = \frac{q}{1-(1-q)^n}$ is convex (the proof for convexity is deferred to the full online version). We want to fit a concave curve U under the convex curve C. Let $\overline{C}_q$ be the tangent to C at q. $\overline{C}_{\hat{q}}$ is a feasible solution for U for all $\hat{q} \in [0,1]$. Further, given that U touches the curve C at $\hat{q}$, $U(t) \leq \overline{C}_{\hat{q}}(t)$ for all $t \in [0,1]$ (from the concavity of U). Also, it is clear that the objective increases with a pointwise increase in U. Thus, we conclude that the optimal solution to the program is achieved at $U = \overline{C}_{\hat{q}}$ at some $\hat{q} \in [0,1]$. In particular, the worst-case ratio occurs when U is a straight-line.

Without loss of generality, we can normalize the slope to 1. Let $U(q) = q + a$. The surplus equals

$$1 + a + \int_0^1 \frac{1-(1-t)^n - nt(1-t)^{n-1}}{t} dt + a \times \int_0^1 \frac{1-(1-t)^n - nt(1-t)^{n-1}}{t^2} dt = H_n + an$$

See the full online version for the proof of the above equality.

By posting a price $v(q)$, we generate a consumer surplus

$$\frac{1-(1-q)^n}{q} \times U(q) = (1 + (1-q) + \cdots + (1-q)^{n-1}) \times (q + a)$$

If $a \geq \frac{1}{n}$, the ratio between the surplus and the consumer surplus is at most $H_n + 1$ by setting $q = 0$. If $a < \frac{1}{n}$, the surplus is less than $H_n + 1$, and by setting $q = 1$, the consumer surplus is at least 1. Thus, the worst-case ratio between surplus and consumer surplus is at most $H_n + 1 = (1 + o(1)) \ln n$.

We now show the result for all distributions. We rewrite the optimal surplus as follows.

$$\int_0^1 v(t) \times n(1-t)^{n-1} = \Big[V(t) \times n(1-t)^{n-1}\Big]_{t=0}^{t=1} + \int_0^1 V(t) \times n(n-1)(1-t)^{n-2} dt$$

The equality follows through integration by parts. From the above expression, it can be concluded that the optimal surplus increases with a pointwise increase in V.

Let $\overline{V}$ be the price-posting surplus curve of the distribution with a price-posting consumer surplus curve $\overline{U}$. $\overline{U}$ is pointwise larger than U, and hence, from Lemma 10, $\overline{V}$ is pointwise larger than V. Hence, the optimal surplus is larger in the distribution with price-posting consumer surplus curve $\overline{U}$. However, from Theorem 5, the consumer surplus from posting prices is identical to both the distributions. Thus, the consumer surplus irregular distribution has a better ratio between surplus and consumer surplus than the consumer surplus regular distribution. Thus, the ratio between optimal surplus to optimal consumer surplus is at most $(1 + o(1)) \ln n$ for all distributions. ◀

6 Conclusion

In this section, we discuss the various advantages and disadvantages of using the reduction framework of Theorem 1. Connecting the pen testing problem to auction theory and building an easy-to-use reduction framework enable us to design near optimal pen testing algorithms for any general feasibility constraint. We are able to construct algorithms that are only a constant factor away from the lower bound described in Qiao and Valiant [23] for various feasibility constraints like matroids ($\gamma(n) = 1$) and knapsack ($\gamma(n) = 2$). However, there is a scope for improvement on the following two fronts:

1. obtaining tight approximation guarantees, and
2. obtaining approximation guarantees under other models of access to the prior distributions.

Tight Approximation Guarantees. The approximation ratio of any pen testing algorithm is at least the optimal omniscient approximation $\zeta(n)$. This follows since the optimal consumer surplus in the underlying auction environment is an upper bound on the performance of the optimal pen testing algorithm. The framework matches this bound up to a multiplicative factor of the approximation $\gamma(n)$ of deferred-acceptance mechanisms. Reducing this gap for various feasibility environments stands as an open problem.

For instance, consider the online environments discussed in the paper, where $\zeta(n) = (1 + o(1)) \ln n$. Our framework yielded an omniscient approximation $2(1 + o(1)) \ln n$ in the oblivious version ($\gamma(n) = 2$) and $\frac{e}{e-1} \cdot (1+o(1)) \ln n$ in the sequential version ($\gamma(n) = \frac{e}{e-1}$). We conjecture that the approximation guarantees obtained by applying the reduction framework are not tight, and can be strengthened to $(1 + o(1)) \ln n$ in both these environments, like in the IID setting (Section 5).

Other Models of Access to Distributions. Our framework crucially makes use of virtual values, which in turn need complete knowledge of the distribution of ink levels. Thus, tailoring this approach to models where the algorithm gets partial access to the priors is challenging.

Qiao and Valiant [23] study two such models. They study the online oblivious problem where the algorithm gets access to just one sample from the distribution of each pen. They give a $O(\log n)$-approximation to the omniscient benchmark in this setting. They also look at the online secretary setting, where the pens arrive according to a permutation chosen uniformly at random. The quantity of ink in the pens are adversarially determined. The algorithm is told the quantity of ink in the pen with the largest ink level. They give a $O(\log n)$-approximation in this environment. Further, when the algorithm is given access to the ink levels in all the n pens (with random arrival order), they tighten their bound to $O(\frac{\log n}{\log \log n})$. It is an open question to identify reduction frameworks for these different models of access to the distributions.

References

1 Saeed Alaei, Hu Fu, Nima Haghpanah, Jason D Hartline, and Azarakhsh Malekian. Bayesian optimal auctions via multi-to single-agent reduction. In *13th ACM Conference on Electronic Commerce, EC'12*, page 17, 2012. doi:10.1145/2229012.2229017.

2 Saeed Alaei, Hu Fu, Nima Haghpanah, Jason D Hartline, and Azarakhsh Malekian. Efficient computation of optimal auctions via reduced forms. *Mathematics of Operations Research*, 44(3):1058–1086, 2019. doi:10.1287/MOOR.2018.0958.

3 Vivi Alatas, Abhijit Banerjee, Rema Hanna, Benjamin A Olken, Ririn Purnamasari, and Matthew Wai-Poi. Ordeal mechanisms in targeting: Theory and evidence from a field experiment in indonesia. Technical report, National Bureau of Economic Research, 2013.

4 Sushil Bikhchandani, Sven De Vries, James Schummer, and Rakesh V Vohra. An ascending vickrey auction for selling bases of a matroid. *Operations research*, 59(2):400–413, 2011. doi:10.1287/OPRE.1100.0888.

5 Mark Braverman, Jing Chen, and Sampath Kannan. Optimal provision-after-wait in healthcare. *Mathematics of Operations Research*, 41(1):352–376, 2016. doi:10.1287/MOOR.2015.0731.

6 Shuchi Chawla, Jason D Hartline, David L Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 311–320, 2010. doi:10.1145/1806689.1806733.

7 Hao Chung and Elaine Shi. Foundations of transaction fee mechanism design. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3856–3899. SIAM, 2023. doi:10.1137/1.9781611977554.CH150.

8 Paul Dütting, Vasilis Gkatzelis, and Tim Roughgarden. The performance of deferred-acceptance auctions. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 187–204, 2014. doi:10.1145/2600057.2602861.

9 Cynthia Dwork, Andrew Goldberg, and Moni Naor. On memory-bound functions for fighting spam. In *Advances in Cryptology-CRYPTO 2003: 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003. Proceedings 23*, pages 426–444. Springer, 2003. doi:10.1007/978-3-540-45146-4_25.

10 Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In *Annual international cryptology conference*, pages 139–147. Springer, 1992. doi:10.1007/3-540-48071-4_10.

11 Michal Feldman, Vasilis Gkatzelis, Nick Gravin, and Daniel Schoepflin. Bayesian and randomized clock auctions. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, pages 820–845, 2022. doi:10.1145/3490486.3538247.

12 Yiding Feng, Jason D Hartline, and Yingkai Li. Simple mechanisms for non-linear agents. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3802–3816. SIAM, 2023. doi:10.1137/1.9781611977554.CH148.

13 Dimitris Fotakis, Dimitris Tsipras, Christos Tzamos, and Emmanouil Zampetakis. Efficient money burning in general domains. *Theory of Computing Systems*, 59:619–640, 2016. doi:10.1007/S00224-016-9720-2.

14 Aadityan Ganesh, Clayton Thomas, and S. Matthew Weinberg. Revisiting the primitives of transaction fee mechanism design. *In Proceedings of the 25th ACM Conference on Economics and Computation (EC '24), July 8–11, 2024, New Haven, CT, USA. ACM, New York, NY, USA, 1 page.*, 2024. doi:10.1145/123456.654321.

15 Kira Goldner and Taylor Lundy. Simple mechanisms for utility maximization: Approximating welfare in the iid unit-demand setting. *arXiv preprint arXiv:2402.12340*, 2024. doi:10.48550/arXiv.2402.12340.

16 Mohammad Taghi Hajiaghayi, Robert Kleinberg, and Tuomas Sandholm. Automated online mechanism design and prophet inequalities. In *AAAI*, volume 7, pages 58–65, 2007. URL: http://www.aaai.org/Library/AAAI/2007/aaai07-009.php.

17 Jason Hartline and Samuel Taggart. Sample complexity for non-truthful mechanisms. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 399–416, 2019. doi:10.1145/3328526.3329632.

18 Jason D Hartline and Tim Roughgarden. Optimal mechanism design and money burning. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 75–84, 2008. doi:10.1145/1374376.1374390.

19 Robert Kleinberg and Seth Matthew Weinberg. Matroid prophet inequalities. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 123–136, 2012.

20 Henrik Jacobsen Kleven and Wojciech Kopczuk. Transfer program complexity and the take-up of social benefits. *American Economic Journal: Economic Policy*, 3(1):54–90, 2011.

21 Paul Milgrom and Ilya Segal. Deferred-acceptance auctions and radio spectrum reallocation. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 185–186, 2014. doi:10.1145/2600057.2602834.

22 Roger B Myerson. Optimal auction design. *Mathematics of operations research*, 6(1):58–73, 1981. doi:10.1287/MOOR.6.1.58.

23 Mingda Qiao and Gregory Valiant. Online pen testing. *14th Innovations in Theoretical Computer Science Conference, (ITCS) 2023, January 10-13, 2023, MIT, Cambridge, USA*, 2023. doi:10.48550/arXiv.2210.00655.

24 Tim Roughgarden. Transaction fee mechanism design. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, EC '21, page 792, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3465456.3467591.

25 Ester Samuel-Cahn. Comparison of threshold stop rules and maximum for independent nonnegative random variables. *Ann. Probab.*, 12(4):1213–1216, 1984. URL: `http://dml.mathdoc.fr/item/1176993150`.

26 Qiqi Yan. Mechanism design via correlation gap. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 710–719. SIAM, 2011. `doi:10.1137/1.9781611973082.56`.

Differential Privacy on Trust Graphs

Badih Ghazi
Google Research, Mountain View, CA, USA

Ravi Kumar
Google Research, Mountain View, CA, USA

Pasin Manurangsi
Google Research, Bangkok, Thailand

Serena Wang
Google Research, Mountain View, CA, USA
Harvard University, Cambridge, MA, USA

Abstract

We study differential privacy (DP) in a multi-party setting where each party only trusts a (known) subset of the other parties with its data. Specifically, given a *trust graph* where vertices correspond to parties and neighbors are mutually trusting, we give a DP algorithm for aggregation with a much better privacy-utility trade-off than in the well-studied local model of DP (where each party trusts no other party). We further study a robust variant where each party trusts all but an unknown subset of at most t of its neighbors (where t is a given parameter), and give an algorithm for this setting. We complement our algorithms with lower bounds, and discuss implications of our work to other tasks in private learning and analytics.

2012 ACM Subject Classification Security and privacy → Information-theoretic techniques; Security and privacy → Trust frameworks; Theory of computation → Computational complexity and cryptography; Theory of computation → Theory of database privacy and security

Keywords and phrases Differential privacy, trust graphs, minimum dominating set, packing number

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.53

Related Version *Full Version*: https://arxiv.org/abs/2410.12045

1 Introduction

Differential privacy (DP) [22, 21] is a rigorous privacy notion that has seen extensive study (e.g., [23, 52]) and widespread adoption in analytics and learning (e.g., [20, 2, 46, 51]). It dictates that the output of a randomized algorithm remains statistically indistinguishable if the data of a single user changes. The most widely studied models of DP are the *central* model where a trusted curator is given access to the raw data and required to output a DP estimate of the function of interest, and the *local* model [26, 37] where every message leaving each user's device is required to be DP. While the latter is compelling in that each user needs to place minimal trust in other users, it is known to suffer from a significantly higher utility degradation compared to the former (e.g., [15]).

In practice, data sharing settings often include situations where a user is willing to place more trust in a *subset* of other users. For example, many people have different privacy sensitivities depending on their relationships with others: Alice might be willing to share her location data with family and close friends, but unwilling to have her location data be recoverable by strangers from a public channel. This relates to philosophical conceptualizations of privacy as *control over personal information*, in that individuals may specify with whom they are willing share their information [49].


© Badih Ghazi, Ravi Kumar, Pasin Manurangsi, and Serena Wang;
licensed under Creative Commons License CC-BY 4.0



Trust Graph DP Models

In this work, we model such relationships as a *trust graph* where vertices correspond to different users and neighboring vertices are mutually trusting (see Figure 1 for a simple example). We define and study DP over such trust graphs, whereby the DP guarantee is enforced on the messages exchanged by each vertex or its trusted neighbors on one hand, and the other non-trusted vertices on the other hand.

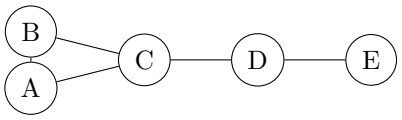


Figure 1 Simple example trust graph. User A is only willing to share their data with users B and C, and user C is additionally willing to share their data with D. We introduce a privacy model (TGDP) in which users D and E cannot identify user A's data based on any communication exchanged.

Specifically, we define notions of *Trust Graph DP (TGDP)* that generalize existing definitions of local DP [37] and central DP, and effectively interpolate between them. Informally, TGDP requires that the distribution of all messages exchanged by each vertex v or one of its neighbors with any vertices that are not trusted by v should remain statistically indistinguishable if the input data held by v changes; we formalize this in Definition 7.

We further extend TGDP to capture *robustness* to potentially compromised neighbors. Namely, the above privacy guarantee can break if a single neighbor of a vertex turns out to be untrustworthy. Thus, we introduce the notion of *Robust Trust Graph DP (RTGDP)*, which maintains the privacy guarantee even if some unknown subset of neighbors of a given size is compromised; see Definition 14.

Our Results

Having defined these trust graph-based models of DP, we give algorithms for the basic *aggregation* primitive that satisfy both TGDP and RTGDP notions. Notably, we propose algorithms that depend on linear programming formulations that can be computed in polynomial time (Theorems 9 and 15). We complement our algorithms with lower bounds on the error that depend on combinatorial properties of the graph (Theorems 10 and 16). Although closing the gap between our upper and lower bounds is still open, we obtain a *bi-criteria* result showing that the upper bound is not much larger than the lower bound when we slightly increase the robustness parameter t (Theorem 17).

The aggregation primitive we study in this work is a basic building block. Indeed, our work implies new DP algorithms over trust graphs for other problems in learning and analytics (see Appendix 5).

We supplement the theory with evaluation on nine real network datasets including email communication networks, social networks, and cryptocurrency trust networks. Our results show that the utility degradation when satisfying TGDP and RTGDP can be significantly lower than that of local DP. Thus, when trust relationships exist, accounting for these relationships when incorporating DP into a system can considerably improve overall utility.

1.1 Technical Overview

We first give a TGDP mechanism for the integer aggregation problem with a mean-squared error (MSE) that scales linearly in the size of any dominating set[1] of the trust graph. The idea of the protocol is the following: Each user identifies one user in the dominating set that they trust and to whom they send their input. Then, each user in the dominating set simply runs the central (discrete) Laplace mechanism to privatize the data. The final estimate is the sum of all these sub-estimates in the dominating set. The MSE grows with the dominating set size, as formalized in Theorem 8.

A disadvantage of the protocol described above is that, in order to minimize the error, it requires the knowledge of a minimum dominating set. However, computing a minimum dominating set is NP-hard and even hard to approximate [27]. So we give (in Theorem 9) a protocol that not only is efficient but can also reduce the error by up to $O(\log n)$ factor. This protocol only requires a solution to a linear program (LP), which, unlike the minimum dominating set, can be computed in polynomial time. At a high-level, there are two key ideas in this protocol:

- **(i)** *Input Splitting:* Instead of sending the input to a single vertex as in the previous section, each user will split their input into (random) additive shares and send it to all its neighbors. The input splitting idea originated in cryptography [36] and has recently found applications in the shuffle model of DP [6, 32], although the nature of how we use it here is quite different from those previous works.
- **(ii)** *Distributed Noise Addition:* Similar to the previous protocol, each user again broadcasts the sum of all messages they receive with some noise added. The main difference here is that, instead of using the discrete Laplace noise, we use the negative binomial noise designed in such a way that, when sufficiently many of them are summed up, they guarantee DP. This helps reduce the amount of noise required in the protocol. (The idea of distributed noise generation dates back to the early works on DP [21], but the distributions we use here are from [6].)

What we gain by applying the input splitting is that, due to the properties of random additive shares, the only way the adversary learns anything about x_v is to sum up all the messages broadcast from its neighbors. By a careful design of the distributed noise distribution, we can ensure that this sum contains sufficient noise to provide DP guarantees.

Our lower bound on integer aggregation with TGDP (Theorem 10) shows that the MSE grows with the packing number of the trust graph. The main idea is to transform any TGDP protocol to a local DP (LDP) protocol with the same privacy and utility, but with a number of users equal to the packing number. The "packing" property ensures that the users are "isolated" from each other in the reduction step.

For our results in the RTGDP model, we consider the same LP but impose a stricter constraint to ensure DP guarantees even when some neighbors of each vertex are compromised. To prove our bi-criteria tightness (Theorem 17), we study the dual of the LP and apply randomized rounding to convert the fractional solution into an integral one.

1.2 Related Work

Secure multiparty computation (SMPC) [55, 56, 34] can be leveraged to allow users to achieve central DP utility without relying on a trusted curator; this is done via cryptographic protocols whose security relies on computational hardness assumptions [21, 8, 11, 9]. An

[1] See Section 3 for the formal definitions of a dominating set and a packing.

important distinction between our model and SMPC is that while the privacy of SMPC protocols relies on computational hardness assumptions, the privacy guaranteed in our proposed TGDP model is information-theoretic (and thus stronger). Still, our proposed TGDP model can also be thought of as relaxing the SMPC threat model to assume that a subset of at least a certain number of users is trustworthy (or would execute the algorithm faithfully). Specifically, the proposed TGDP notion could enable higher-utility protocols than the state-of-the-art in SMPC, by making a stronger (but realistic) assumption that different users fully trust some subset of other users (namely, their neighbors in the trust graph).

We also point out that the multi-central (a.k.a. multi-server) model of DP [50, 17] is a special case of our RTGDP model. Another intermediate model between local and central DP is the shuffle model [10, 25, 16], where aggregation has been extensively studied (e.g., [5, 30, 31]); but this model does not capture mutually trusting relationships between different pairs of users. Finally, the network DP model [18] is a different relaxation of local DP where the (DP) communication between users is restricted to the edges of a given graph; this is in contrast to our proposed TGDP model where a given graph encodes trust relationships.

Organization

We start with some background in Section 2. In Section 3, we formally define the notion of DP on trust graphs, and give algorithms and a lower bound for solving the aggregation task under this privacy guarantee. The RTGDP notion is defined in Section 4 where we also give an algorithm and lower bounds for aggregation under this robust notion. We conclude with some interesting future directions in Section 6. Missing proofs are in Appendix B. In Appendix C, we include experiments in which we report our given upper and lower bounds on real network datasets.

2 Preliminaries

Let n users be represented as vertices $V = \{1, ..., n\}$ of a graph $G = (V, E)$, where $E \subseteq V^2$ corresponds to the set of pairs (i, j) of users, where i and j are willing to share their data with each other.[2] Let $\mathcal{X}$ be any domain and suppose each user $i \in V$ has data $x_i \in \mathcal{X}$, and the (full) input dataset is given by $(x_1, \ldots, x_n) = \mathbf{x} \in \mathcal{X}^n$. Let $N(v)$ be the *neighborhood* of v in $G = (V, E)$, i.e., $N(v) = \{u \mid (u, v) \in E\}$ and let $N[v]$ be the *closed neighborhood* of v, i.e., $N[v] = N(v) \cup \{v\}$.

2.1 Differential Privacy Definitions and Tools

▶ **Definition 1** (DP; [22, 21]). *A randomized mechanism $M : \mathcal{X}^n \to \mathcal{O}$ is* (ε, δ)-differentially private ((ϵ, δ)-DP) *if for all pairs $\mathbf{x}, \mathbf{x}' \in \mathcal{X}^n$ of datasets that differ only in the data of a single user, and for all subsets $S \subseteq \mathcal{O}$,* $\Pr[M(\mathbf{x}) \in S] \leq e^{\varepsilon} \Pr[M(\mathbf{x}') \in S] + \delta$.

For brevity, we write $(\varepsilon, 0)$-DP as ε-DP (a.k.a., *pure*-DP).

In *non-interactive local DP*, each user has to randomize their own input and send it to the server (or, alternatively, publish it). In this case, each user's randomized output is required to be DP:

[2] For simplicity, we focus only on the "symmetric" notion of sharing. It is relatively simple to extend all of our algorithms to the "asymmetric" version as well.

▶ **Definition 2** (Non-Interactive Local DP; [37]). *A randomized mechanism $M : \mathcal{X} \to \mathcal{O}$ is a* non-interactive (ε, δ)-local DP randomizer *if for any pair $x, x' \in \mathcal{X}$, and for all subsets $S \subseteq \mathcal{O}$, $\Pr[M(x) \in S] \leq e^{\varepsilon} \Pr[M(x') \in S] + \delta$.*

To define DP properties of possibly interactive protocols, we follow the approach of [8]. First, we define the notion of a protocol view.

▶ **Definition 3** (View). *The* view *of a protocol P at vertex u for an input dataset $\mathbf{x} \in \mathcal{X}^n$, denoted $\textsc{view}_P^u(\mathbf{x})$, consists of the input x_u and all messages received and sent (together with the corresponding source/destination) by the vertex u.*

The view of the protocol P for a subset $S \subseteq V$ of vertices is defined as $\textsc{view}_P^S(\mathbf{x}) := (\textsc{view}_P^u(\mathbf{x}))_{u \in S}$. Let $\mathcal{O}$ be the set of all possible views. For any $T \subseteq V$, we write $\mathbf{x}_{-T}$ as a shorthand for $\mathbf{x}_{V \setminus T}$ and, for any $v \in V$, $\mathbf{x}_{-v}$ as a shorthand for $\mathbf{x}_{-\{v\}}$. When the protocol is interactive, local DP can be defined as follows [8]:

▶ **Definition 4** (Interactive Local DP). *A protocol P satisfies (ϵ, δ)-*local DP *((ϵ, δ)-LDP) if for each vertex $v \in V$, $\textsc{view}_P^{V \setminus \{v\}}(\mathbf{x})$ satisfies (ε, δ)-DP with respect to the input x_v for all values of $\mathbf{x}_{-v}$. I.e., for all pairs $x_v, x_v' \in \mathcal{X}$, all values of $\mathbf{x}_{-v}$, and all subsets $S \subseteq \mathcal{O}$,*

$$\Pr[\textsc{view}_P^{V \setminus \{v\}}(x_v, \mathbf{x}_{-v}) \in S] \leq e^{\varepsilon} \Pr[\textsc{view}_P^{V \setminus \{v\}}(x_v', \mathbf{x}_{-v}) \in S] + \delta.$$

For pure-DP, it is useful to define $D_\infty(\mathcal{P} \parallel \mathcal{P}') := \max_{o \in \mathrm{supp}(\mathcal{P})} \ln\left(\frac{\Pr_{X \sim \mathcal{P}}[X=o]}{\Pr_{X' \sim \mathcal{P}}[X'=o]}\right)$ for distributions $\mathcal{P}, \mathcal{P}'$. We will sometimes use random variables and distributions interchangeably.

It is well-known that DP is robust to post-processing. This fact will be useful in our privacy analysis.

▶ **Lemma 5** (Post-Processing). *For any random variables X, X' and a (possibly randomized) function f, we have $D_\infty(f(X) \parallel f(X')) \leq D_\infty(X \parallel X')$.*

We will use the following distributions for the noise:

- The *negative binomial distribution* $\mathrm{NB}(r, p)$ with parameters $r > 0, p \in (0, 1)$ is supported on $\mathbb{Z}_{\geq 0}$ with density $\Pr[X = k] = \binom{k+r-1}{k}(1-p)^k p^r$, where $X \sim \mathrm{NB}(r, p)$. Its variance is $r(1-p)/p^2$.
- Let $\mathrm{sNB}(r, p)$ be the distribution of $X - X'$ where $X, X' \sim \mathrm{NB}(r, p)$ are i.i.d.
- The *discrete Laplace distribution* $\mathrm{DLap}(b)$ with parameter $b > 0$ is supported on $\mathbb{Z}$ and its density is given by $\Pr[X = k] \propto \exp(-|k|/b)$ for $X \sim \mathrm{DLap}(b)$.

We will use the following facts in our analysis:

- If $X_1 \sim \mathrm{sNB}(r_1, p)$ and $X_2 \sim \mathrm{sNB}(r_2, p)$, then $X_1 + X_2 \sim \mathrm{sNB}(r_1 + r_2, p)$.
- $\mathrm{DLap}(b)$ is the same distribution as $\mathrm{sNB}(1, 1 - e^{-1/b})$.

The discrete Laplace mechanism is well-known to guarantee DP in the central setting [33]. Below, we state a slightly more general version of this for sNB that will be convenient for our analysis.

▶ **Lemma 6.** *For any $x, x' \in \{0, \ldots, \Delta\}$, let $Z \sim \mathrm{sNB}(r, 1 - e^{-\varepsilon/\Delta})$ where $r \geq 1$. Then, we have $D_\infty(Z + x \parallel Z + x') \leq \varepsilon$.*

In the privacy analysis, we often consider $\textsc{view}_P^S(x_v, \mathbf{x}_{-v})$ and $\textsc{view}_P^S(x_v', \mathbf{x}_{-v})$ for $x_v, x_v' \in \mathcal{X}$. For convenience, we will write $\textsc{view}_P^S(x)$ for $x \in \mathcal{X}$ as a shorthand for $\textsc{view}_P^S(x_v, \mathbf{x}_{-v})$ when $x_v = x$. Similarly, for a quantity y that depends on x_v, we will write $y(x)$ to denote y when $x = x_v$.

3 Trust Graph Differential Privacy

We model trust relationships across users as a network where vertices correspond to users, and undirected edges connect users who are mutually willing to share their data (see Figure 1). We focus on undirected trust graphs, though extensions of our results to directed graphs are possible. For a given trust graph, we define a general notion of Trust Graph DP, provide algorithms for achieving it, and analyze upper and lower bounds on the error for the integer aggregation problem.

▶ **Definition 7.** *(Trust Graph DP) Let $G = (V, E)$. A protocol P satisfies* (ϵ, δ, G)-Trust Graph DP *((ϵ, δ, G)-TGDP) if for each vertex $v \in V$, $\textsc{view}_P^{V \setminus N[v]}(\mathbf{x})$ satisfies (ε, δ)-DP with respect to the input x_v for all values of $\mathbf{x}_{-v}$. I.e., for all pairs $x_v, x'_v \in \mathcal{X}$, all values of $\mathbf{x}_{-v}$, and all subsets $S \subseteq \mathcal{O}$,*

$$\Pr[\textit{VIEW}_P^{V \setminus N[v]}(x_v, \mathbf{x}_{-v}) \in S] \leq e^{\varepsilon} \Pr[\textit{VIEW}_P^{V \setminus N[v]}(x'_v, \mathbf{x}_{-v}) \in S] + \delta.$$

Referring back to Figure 1 as an example, Definition 7 says that even if users D and E pooled their messages together, their collective view would still be DP with respect to the data for user A.

Notably, the proposed TGDP model generalizes both the central DP model and the local DP model. The central DP model is captured when G is a star graph in which all vertices (the users) entrust their data a single central vertex (the analyst): each user's data is private relative to the view of all other users, but all users trust the same central analyst. The local DP model, on the other hand, is captured when G simply has no edges between any vertices. Our TGDP model thus introduces a flexibility to capture intermediate trust relationships, perhaps involving several local analysts, or more general trust graphs arising from social networks.

As before, we write $(\varepsilon, 0, G)$-TGDP as (ε, G)-TGDP. Note that the (ε, G)-DP condition in Definition 7 can be written as $D_\infty\left(\textsc{view}_P^{V \setminus N[v]}(x_v) \,\middle\|\, \textsc{view}_P^{V \setminus N[v]}(x'_v)\right) \leq \varepsilon$.

Recall that for a graph $G = (V, E)$, a *dominating set* is a subset $U \subseteq V$ such that for every $v \in V \setminus U$, there is a $u \in U$ such that $(u, v) \in E$; the size of a minimum dominating set is the *domination number* $\gamma(G)$. A *packing* of G is a subset $U \subseteq V$ such that for any distinct $u, u' \in U$, $N[u]$ and $N[u']$ are disjoint; the size of a maximum packing is the *packing number* $\rho(G)$.

Aggregation

We consider the *integer aggregation* problem. Let each individual have a value $x_i \in \{0, \ldots, \Delta\}$. The goal is to compute an estimate $\tilde{a}$ of $a = \sum_{i=1}^{n} x_i$. We measure the *mean-square error* (MSE), which is defined as $\mathbb{E}[(\tilde{a} - a)^2]$, where the expectation is over the randomness of the protocol. In central DP, the standard Laplace mechanism [22] achieves an error of $2\Delta^2/\varepsilon^2$. In local DP, the local version of the Laplace mechanism achieves an error of $2\Delta^2 n/\varepsilon^2$. Both of these are known to be asymptotically optimal.

3.1 Algorithm via Dominating Set

We start by giving a protocol for the integer aggregation problem using the graph's dominating set.

▶ **Theorem 8.** *There is an (ε, G)-TGDP mechanism for the aggregation problem with MSE at most $2\Delta^2|T|/\varepsilon^2$, where T is any dominating set of G.*

Proof. The protocol works as follows:

- First, each user $v \in V$ picks an arbitrary vertex $u_v \in T \cap N[v]$. (The intersection is not empty since T is a dominating set.) Then, the user sends x_v to u_v.
- Each user $u \in T$ broadcasts the sum of all numbers it receives together with a noise drawn from $\mathrm{DLap}(\Delta/\varepsilon)$. More formally, the user broadcasts $a_u = \sum_{\substack{v \in V \\ u_v = u}} x_v + z_u$, where $z_u \sim \mathrm{DLap}(\Delta/\varepsilon)$.
- Finally, the estimate is $\tilde{a} = \sum_{u \in T} a_u$.

Privacy Analysis. Consider any $v \in V$ and $\mathbf{x}_{-v} \in \{0, \ldots, \Delta\}^{V \setminus \{v\}}$. We write $\mathbf{a}$ as a shorthand for $(a_u)_{u \in T}$. Let $S_v := \{w \in N[v] \mid u_w \in N[v]\}$ denote the nodes in $N[v]$ whose message in the first step is sent to a node in $N[v]$. Notice that $\mathrm{VIEW}_P^{V \setminus N[v]}(x)$ is exactly $(\mathbf{x}_{-S_v}, \mathbf{a}(x))$. We claim that this is a post-processing of $z_{u_v} + x$. This is simply because $\mathbf{x}_{-S_v}, (a_u)_{u \in T \setminus \{u_v\}}$ do not depend on $x_v = x$ at all and are independent of $z_{u_v} + x$; finally, note that $a_{u_v}(x)$ is a post-processing of $z_{u_v} + x$ since $a_{u_v}(x) = (z_{u_v} + x) + \sum_{\substack{v' \in V \setminus \{v\} \\ u_{v'} = u_v}} x_{v'}$.

Consider any $x_v, x'_v \in \{0, \ldots, \Delta\}$. By Lemma 5 and Lemma 6, we have

$$D_\infty \left(\mathrm{VIEW}_P^{V \setminus N[v]}(x_v) \,\middle\|\, \mathrm{VIEW}_P^{V \setminus N[v]}(x'_v) \right) \le D_\infty(z_{u_v} + x_v \parallel z_{u_v} + x'_v) \le \varepsilon,$$

where the second inequality is due to Lemma 6. Thus, the protocol satisfies (ε, G)-TGDP as desired.

Utility Analysis. The MSE is $\mathbb{E}\left[(\tilde{a} - a)^2\right] = \sum_{u \in T} \mathbb{E}[z_u^2] \le |T| \cdot \frac{2\Delta^2}{\varepsilon^2}$. ◀

3.2 Improved Algorithm via Linear Programming

A disadvantage of the protocol from Section 3.1 is that to minimize the error, it requires the knowledge of a minimum dominating set. Computing minimum dominating set is NP-hard and even hard to approximate [27]. In this section, we give a protocol that is efficient to compute and furthermore can reduce the error by up to $O(\log n)$ factor in certain graphs. To describe our protocol, recall the linear programming (LP) relaxation of the dominating set problem:

$$\min \sum_{u \in V} y_u \qquad \text{s.t.} \sum_{u \in N[v]} y_u \ge 1 \quad \forall v \in V; \qquad 0 \le y_u \le 1 \quad \forall u \in V. \tag{1}$$

To see that this is a relaxation of the dominating set problem, note that any dominating set $T \subseteq V$ gives a solution by setting $y_v = \mathbf{1}[v \in T]$. Due to this, the optimum of this LP is no more than the size of the dominating set. In fact, the LP optimum can be smaller than the minimum dominating set size by an $O(\log n)$ factor [45].

The main result is a protocol whose MSE scales with the LP optimum instead of dominating set:

▶ **Theorem 9.** *There is an (ε, G)-TGDP mechanism for the aggregation problem with MSE at most $2\Delta^2 \cdot \mathrm{OPT}_{\mathrm{LP}}/\varepsilon^2$, where $\mathrm{OPT}_{\mathrm{LP}}$ denotes the value of the optimal solution to the LP in (1).*

Proof. Let $\mathbf{y} = (y_u)_{u \in V}$ denote any solution to the LP in (1). The protocol works as follows:

- Let $q = 2n\Delta$.

- For every user $v \in V$, pick $\{s_v^u\}_{u \in N[v]} \subseteq \mathbb{Z}_q$ uniformly at random among those that satisfy $\sum_{u \in N[v]} s_v^u \equiv x_v \mod q$. Then, for every $u \in N[v]$, user v sends s_v^u to u.
- For every $u \in V$, sample $z_u \sim \mathrm{sNB}(y_u, 1 - e^{-\varepsilon/\Delta})$; broadcast $a_u \equiv z_u + \sum_{v \in N[u]} s_v^u \mod q$.
- Compute $a' \equiv \sum_u a_u \mod q$. Then, output $\tilde{a} = \begin{cases} a' & \text{if } a' \le q/2, \\ a' - q & \text{otherwise.} \end{cases}$

Privacy Analysis. Throughout the analysis, we assume that the addition is modulo q unless stated otherwise. Consider any $v \in V$ and $\mathbf{x}_{-v} \in \{0, \dots, \Delta\}^{V \setminus \{v\}}$. We write $\mathbf{a}$ as a shorthand for $(a_u)_{u \in V}$. Notice that $\mathrm{VIEW}_P^{V \setminus N[v]}(x)$ is exactly $(\mathbf{x}_{-N[v]}, \mathbf{a}(x), (s_{v'}^u)_{u \in V \setminus N[v], v' \in V})$. We claim that this is a post-processing of $(z_u + s_v^u)_{u \in N[v]}$. This is simply because $\mathbf{x}_{-N[v]}, (a_u)_{u \in V \setminus N[v]}, (s_{v'}^u)_{u \in V \setminus N[v], v' \in V}$ do not depend on $x_v = x$ at all and are independent of $(z_u + s_v^u)_{u \in N[v]}$; finally, note that $(a_u(x))_{u \in N[v]}$ is a post-processing of $(z_u + s_v^u)_{u \in N[v]}$ since $a_u(x) = (z_u + s_v^u) + \sum_{\substack{v' \in N[v] \setminus \{v\} \\ u_{v'} = u}} s_{v'}^u$ for all $u \in N[v]$.

For any $x_v, x_v' \in \{0, \dots, \Delta\}$, Lemma 5 implies that

$$D_\infty \left(\mathrm{VIEW}_P^{V \setminus N[v]}(x_v) \,\middle\|\, \mathrm{VIEW}_P^{V \setminus N[v]}(x_v') \right) \le D_\infty((z_u + s_v^u(x_v))_{u \in N[v]} \,\|\, (z_u + s_v^u(x_v'))_{u \in N[v]}).$$

Now, since $(s_v^u(x))_{u \in N[v]}$ are random elements of $\mathbb{Z}_q$ that sum to x, we also have that $(z_u + s_v^u(x))_{u \in N[v]}$ are random elements of $\mathbb{Z}_q$ that sum to $x + \sum_{u \in N[v]} z_u$. In other words, $(z_u + s_v^u(x_v))_{u \in N[v]}$ is a post-processing of $x + \sum_{u \in N[v]} z_u$. Again, Lemma 5 implies that

$$D_\infty((z_u + s_v^u(x_v))_{u \in N[v]} \,\|\, (z_u + s_v^u(x_v'))_{u \in N[v]}) \le D_\infty \left(x_v + \sum_{u \in N[v]} z_u \,\middle\|\, x_v' + \sum_{u \in N[v]} z_u \right).$$

Finally, $Z := \sum_{u \in N[v]} z_u$ is distributed as $\mathrm{sNB}\left(\sum_{u \in N[v]} y_u, 1 - e^{-\varepsilon/\Delta}\right)$. Since $\mathbf{y}$ is feasible in (1), we have $\sum_{u \in N[v]} y_u \ge 1$. Thus, we can apply Lemma 6 to conclude that the RHS above is $\le \varepsilon$.

Utility Analysis. Note $\tilde{a} \equiv a + \left(\sum_{u \in V} z_u\right) \mod q$. Since $a \in [0, q/2]$ and $\tilde{a} \in (-q/2, q/2]$, we have $|a - \tilde{a}| \le \left|\sum_{u \in V} z_u\right|$. Thus, the MSE is $\le \sum_{u \in T} \mathbb{E}[z_u^2] \le \sum_{u \in T} \frac{2y_u}{(\varepsilon/\Delta)^2} = \frac{2\Delta^2 \mathrm{OPT}_{\mathrm{LP}}}{\varepsilon^2}$, where the last equality is from our assumption that $(y_u)_{u \in V}$ is an optimal solution to the LP in (1). ◀

We remark that, in the proof above, the privacy guarantee holds even for non-optimal LP solution $\mathbf{y}$, as long as it satisfies the constraints. Similarly, the error guarantee holds where $\mathrm{OPT}_{\mathrm{LP}}$ is replaced with the objective value of the solution. This is helpful for practical applications where we may only have an approximately optimal LP solution.

3.3 Lower Bound

We now give a lower bound for integer aggregation, where the MSE grows with the packing number.

▶ **Theorem 10.** *For any $\varepsilon \le O(1)$, any (ε, G)-TGDP protocol for integer aggregation incurs MSE $\Omega(\Delta^2 \cdot \rho(G))$, where $\rho(G)$ denotes the packing number of the trust graph G.*

In fact, we give the following reduction that transforms any TGDP protocol to an LDP protocol with the same privacy parameter and MSE, but only on $\rho(G)$ users (instead of n users). Applying the known $\Omega(\Delta^2 n)$ lower bound for integer aggregation in LDP [15][3] immediately yields Theorem 10.

▶ **Lemma 11.** *Suppose that there is an (ε, G)-TGDP protocol for integer aggregation. Then, there exists an ε-local DP protocol for integer aggregation for $\rho(G)$ users with the same MSE as the (ε, G)-TGDP protocol, where $\rho(G)$ denotes the packing number of G.*

Proof. Let $U = \{u_1, \ldots, u_m\} \subseteq V$ be the largest packing in G where $m = \rho(G)$. To avoid ambiguity, let $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_m)$ be the input to the LDP protocol (that we construct below).

To construct the LDP protocol, let $Q_1 \cup \cdots \cup Q_m$ be any partition of V such that $N[u_i] \subseteq Q_i$ for all $i \in [m]$. Such a partition exists because $N[u_1], \ldots, N[u_m]$ are disjoint by the definition of packing. Let P be any (ε, G)-TGDP protocol for integer aggregation. Our LDP protocol $\tilde{P}$ runs the protocol P where each $\tilde{P}$'s user $i \in [m]$ assumes the role of all P's users in Q_i, where the input to P is defined as $x_u = \begin{cases} \tilde{x}_i & \text{if } u = u_i \\ 0 & \text{otherwise,} \end{cases}$ $\forall u \in Q_i$. We then output the estimate as produced by P. The MSE of $\tilde{P}$ is obviously the same as that of P.

To see that $\tilde{P}$ satisfies ε-LDP, consider any $i \in [m], \tilde{\mathbf{x}}_{-i} \in \mathcal{X}^{[m] \setminus \{i\}}$, we have $\textsc{view}_{\tilde{P}}^{[m] \setminus \{i\}}(\tilde{x}) = \textsc{view}_P^{V \setminus Q_i}(\mathbf{x}(\tilde{x}))$, where $\mathbf{x}(\tilde{x})$ is the input to P as defined above. Since $V \setminus Q_i \subseteq V \setminus N[u_i]$, $\textsc{view}_P^{V \setminus Q_i}(\mathbf{x}(\tilde{x}))$ is a post-processing of $\textsc{view}_P^{V \setminus N[u_i]}(\mathbf{x}(\tilde{x}))$, Lemma 5 implies that

$$D_\infty\left(\textsc{view}_{\tilde{P}}^{[m] \setminus \{i\}}(\tilde{x}_i) \,\middle\|\, \textsc{view}_{\tilde{P}}^{[m] \setminus \{i\}}(\tilde{x}_i')\right) \le D_\infty\left(\textsc{view}_P^{V \setminus N[u_i]}(\tilde{x}_i) \,\middle\|\, \textsc{view}_P^{V \setminus N[u_i]}(\tilde{x}_i')\right) \le \varepsilon,$$

where the last inequality is due to P being an (ε, G)-TGDP protocol. Hence, $\tilde{P}$ is ε-LDP. ◀

Unfortunately, the lower bound in Theorem 10 is not tight with respect to the upper bounds in Theorems 8 and 9. Indeed, the following is a example of a graph that has a large gap between the domination number and the packing number [14]. Let $V = [k] \times [k]$ for $k \in \mathbb{N}$. There is an edge between any $(x, y) \in V, (x', y') \in V$ iff $x = x'$ or $y = y'$. For this graph, $\mathrm{OPT}_{\mathrm{LP}}$ is$\Omega(\sqrt{|V|})$ since every vertex has degree $O(k) = O(\sqrt{|V|})$ whereas the maximal packing has size exactly one (see Figure 2 for $k = 4$). We can also show that this instance exhibits an asymptotically optimal gap:

▶ **Theorem 12.** *For any graph G, $\mathrm{OPT}_{\mathrm{LP}} \le \rho(G) \cdot \sqrt{n}$ where $\mathrm{OPT}_{\mathrm{LP}}$ denote the value of the optimal solution to the LP in (1).*

In other words, our upper bound based on the LP (Theorem 9) and our lower bound (Theorem 10) on the MSE has a gap of at most $O(\sqrt{n})$. To the best of our knowledge, the bound in Theorem 12 was not known before; we give the full proof in Section D.

Note that the above instance also gives a gap of $\Omega(\sqrt{|V|})$ between the domination number and the packing number. Since it is known [45] that $\gamma(G) \le O(\log n) \cdot \mathrm{OPT}_{\mathrm{LP}}$, Theorem 12 implies the following corollary:

▶ **Corollary 13.** *For any graph G, $\gamma(G) \le \rho(G) \cdot O(\sqrt{n} \cdot \log n)$.*

That is, the above gap instance is tight up to a logarithmic factor. Furthermore, this also means that our upper bound based on the dominating set (Theorem 8) and our lower bound (Theorem 10) on the MSE has a gap of at most $O(\sqrt{n})$.

[3] Note that [15] state their lower bound for $\Delta = 1$ but the case $\Delta > 1$ follows by scaling up the input.

4 Robust Trust Graph Differential Privacy

In the previous section, we assumed that each user u trusts all of their neighbors $N(u)$. Although this is certainly a reasonable assumption, it might pose a security risk. For example, if one of the neighbors of u is compromised, then u's data might be leaked as the model offers no protection with respect to the view of u's neighbors. Indeed, in the dominating set protocol (Theorem 8), the user sends their raw data to one of their neighbors; if this neighbor is compromised, then the user's data is leaked in the clear. To mitigate this, we propose a revised trust graph DP model that is more robust to such leakage. In particular, for each user u, the DP protection remains as long as at most t_u of their neighbors are compromised, where t_u is some predefined number. This is formalized below.

▶ **Definition 14.** *(Robust Trust Graph DP) Let $G = (V, E)$ and $\mathbf{t} = (t_v)_{v \in V} \in \mathbb{Z}^V_{\geq 0}$. A protocol P satisfies $(\epsilon, \delta, G, \mathbf{t})$*-Robust Trust Graph DP ($(\epsilon, \delta, G, \mathbf{t})$-RTGDP) *if for each vertex $v \in V$ and every set $T \subseteq N(v)$ of size at most t_v, $\textsc{view}_P^{V \setminus (N[v] \setminus T)}(\mathbf{x})$ satisfies (ε, δ)-DP with respect to the input x_v for all values of $\mathbf{x}_{-v}$. I.e., for all pairs $x_v, x'_v \in \mathcal{X}$, all values of $\mathbf{x}_{-v}$, and all subsets $S \subseteq \mathcal{O}$,*

$$\Pr[\textsc{view}_P^{V \setminus (N[v] \setminus T)}(x_v, \mathbf{x}_{-v}) \in S] \leq e^{\varepsilon} \Pr[\textsc{view}_P^{V \setminus (N[v] \setminus T)}(x'_v, \mathbf{x}_{-v}) \in S] + \delta.$$

4.1 Integer Aggregation Protocol

We start by giving an integer aggregation protocol that is again based on an LP. We adapt the LP in (1) by imposing a stricter constraint to ensure DP guarantees even when up to t_v of v's neighbor are compromised. This results in the following LP where the only difference compared to (1) is the stricter first constraint.[4] Note that when $\mathbf{t} = 0$, the two LPs coincide.

$$\min \sum_{u \in V} y_u \qquad \text{s.t.} \sum_{u \in (N[v] \setminus T)} y_u \geq 1 \quad \forall v \in V,\ T \in \binom{N(v)}{\leq t_v}; \quad 0 \leq y_u \leq 1 \quad \forall u \in V. \tag{2}$$

While (2) can have exponential size due to the presence of "$\forall T \in \binom{N(v)}{\leq t_v}$" in the constraint, it can still be solved in polynomial time because an efficient separation oracle exists for the first constraint. Specifically, for each $v \in V$, we can check the first constraint by letting T be the t_v maximum values and check the inequality for just that T (instead of enumerating over all $T \in \binom{N(v)}{\leq t_v}$). From the above LP, we can derive an algorithm that uses exactly the same protocol as in Theorem 9.

▶ **Theorem 15.** *There is an $(\varepsilon, G, \mathbf{t})$-RTGDP protocol for the aggregation problem with MSE at most $2\Delta^2 \cdot \mathrm{OPT}^{\mathbf{t}}_{\mathrm{LP}}/\varepsilon^2$, where $\mathrm{OPT}^{\mathbf{t}}_{\mathrm{LP}}$ denotes the value of the optimal solution to the LP in (2).*

4.2 Lower Bound

We define a $(2, \mathbf{t})$-robust packing of $G = (V, E)$ as a pair $U \subseteq V$ and $(T_u)_{u \in U}$ such that (i) $N[u] \setminus T_u$ are disjoint for all $u \in U$ and (ii) $T_u \in \binom{N(u)}{\leq t_u}$ for all $u \in U$. The size of the robust packing is $|U|$. Let $\rho(G)^{\mathbf{t}}$ denote the largest size of a $(2, \mathbf{t})$-robust packing of G.

[4] $\binom{S}{\leq t}$ denotes the collection of all subsets of S of size $\leq t$.

Note that, when $\mathbf{t} = \mathbf{0}$, $(2, \mathbf{t})$-robust packing coincides with the standard notion of packing we used in the previous section. We prove a lower bound for integer aggregation in the $(\varepsilon, G, \mathbf{t})$-RTGDP model that grows with the size of the maximum $(2, \mathbf{t})$-robust packing:

▶ **Theorem 16.** *For any $\varepsilon \leq O(1)$ and $\mathbf{t} \in \mathbb{N}^V$, any $(\varepsilon, G, \mathbf{t})$-RTGDP protocol for integer aggregation must incur MSE at least $\Omega(\Delta^2 \cdot \rho(G)^{\mathbf{t}})$.*[5]

4.3 Bi-criteria Tightness of the Bounds

Although we are not aware in general how large the gap between our upper (Theorem 15) and lower bounds (Theorem 16) are, we can show the following *bi-criteria* result, that the upper bound is not much larger than the lower bound when we increase $\mathbf{t}$ slightly. This is stated and proved below.

We write $\lceil \alpha \cdot \mathbf{t} \rceil$ for some $\alpha > 0$ as a shorthand for the vector $(\lceil \alpha \cdot t_u \rceil)_{u \in U}$. Furthermore, let $\deg_U$ denote the vector of degrees of the vertices, i.e., $(\deg(u))_{u \in U}$.

▶ **Theorem 17.** *For any $\alpha \in (0, 1)$, $\rho(G)^{\mathbf{t} + \lceil \alpha \cdot \deg_U \rceil} \geq \frac{\alpha}{8} \cdot \mathrm{OPT}^{\mathbf{t}}_{\mathrm{LP}}$, where $\rho(G)^{\mathbf{t}}, \mathrm{OPT}^{\mathbf{t}}_{\mathrm{LP}}$ are as defined in Theorems 15 and 16.*

To show this, we consider the dual of LP in (2), which turns out to be a relaxation for $(2, \mathbf{t})$-robust packing where there is a variable $w_{v,T} \in [0, 1]$ for all $v \in V, T \subseteq \binom{N(v)}{\leq t_v}$ representing whether (v, T) should be included in the robust packing. To turn such a fractional solution to an integral one, we employ *randomized rounding* – a standard technique in approximation algorithms (e.g., [54, Chapter 5]). More precisely, we include (v, T) in our solution with probability proportional to $w_{v,T}$. Unfortunately, this does not work yet as the produced solution may not be a $(2, \mathbf{t})$-robust packing, i.e., it might contain $w_{v,T}$ and $w_{v',T'}$ such that $(N[v] \setminus T)$ and $(N[v'] \setminus T')$ are not disjoint. Due to this, we need to apply a correction procedure on top of this randomized solution. Roughly speaking, we try to enlarge T until we are sure that such an intersection is avoided. This is indeed the reason why we need the slight increase in $\mathbf{t}$. However, even with this increase, we still have to be careful as sometimes T might become too large, i.e., larger than $t_v + \lceil \alpha \cdot \deg(v) \rceil$. We deal with this by simply removing such a pair (v, T) from the solution. A careful analysis shows that (in expectation) only a small fraction of the solution will get removed this way; see Section B.

5 Machine Learning with Trust Graph DP

While the main body of our work focuses on integer aggregation, it is a primitive on which we can build many more complex algorithms. First of all, we can easily use it to perform *real number aggregation* by re-scaling and discretization [6]. In particular, suppose that each user now has $x_i \in [0, 1]$. They can pick $\Delta \in \mathbb{N}$ and perform integer aggregation on $y_i = \mathrm{round}(\Delta x_i)$ where $\mathrm{round}(x)$ randomly round x to either $\lfloor x \rfloor$ or $1 + \lfloor x \rfloor$ with probabilities $1 - (x - \lfloor x \rfloor)$ and $x - \lfloor x \rfloor$ respectively. Once we have run the integer aggregation protocol, the answer is scaled by a factor of $\frac{1}{\Delta}$. If our integer aggregation protocol has MSE $\Delta^2 \xi^2$, then this results in a real aggregation protocol with error $\xi^2 + \frac{n}{4\Delta^2}$ [6]. Picking Δ to be sufficiently large (e.g., $\omega(\sqrt{n})$), the second term becomes negligible.

Real number aggregation allows us to perform statistical queries (SQ) [38]. While a simple family, statistical queries have wide variety of applications in learning theory. One specific work we wish to highlight is that of [29], who showed that convex optimization problems can be solved using statistical queries. As a result, we can apply their algorithm

[5] Again, this theorem is shown via a reduction to the LDP model; see Section B for the proof.

to our setting and obtain convex optimization algorithms with Trust Graph DP. We also remark that statistical queries are also useful for (non-ML) data analytic tasks; e.g., [29] provides SQ-based (vector) mean estimation, and other statistics such as quantiles are also known to be computable via SQs [28].

5.1 Vector Summation with Trust Graph DP

Although SQ-based algorithms can be used in our model, we will sketch a more direct algorithm for the task of *vector summation* with Trust Graph DP. This is not only more efficient but also provide better error guarantees for subsequent tasks such as convex empirical risk minimization (ERM).

In the *vector summation* (with ℓ_2-norm bound) problem, each user input x_i is a vector in $\mathbb{R}^d$ such that $\|x_i\|_2 \leq \Delta$, where Δ is a norm bound known to the algorithm. The goal is again to compute an estimate $\tilde{a}$ to the sum $a = \sum_{i\in[n]} x_i$. The ℓ_2^2-error is defined as $\mathbb{E}[\|\tilde{a} - a\|_2^2]$. Furthermore, we say that the estimator is unbiased if $\mathbb{E}[\tilde{a}] = a$.

It will be easiest to state the algorithms in terms of *zero-concentrated differential privacy (zCDP)* [13, 24]. To do so, we first define α-Renyi divergence for $\alpha > 1$ between distributions $\mathcal{P}, \mathcal{P}'$ for two distributions $\mathcal{P}, \mathcal{P}'$ to be $D_\alpha(\mathcal{P} \parallel \mathcal{P}') := \frac{1}{\alpha-1} \ln \left(\mathbb{E}_{x\sim\mathcal{P}} \left[\left(\frac{\mathcal{P}(x)}{\mathcal{P}'(x)} \right)^{\alpha-1} \right] \right)$. We note that $\lim_{\alpha\to\infty} D_\alpha(\mathcal{P} \parallel \mathcal{P}')$ is indeed equal to $D_\infty(\mathcal{P} \parallel \mathcal{P}')$ that we defined in Section 2.

zCDP can now be defined as follows.

▶ **Definition 18** (zCDP; [13]). *A randomized mechanism $M : \mathcal{X}^n \to \mathcal{O}$ is* ρ-zero concentrated DP (ρ-zCDP) *if for all pairs $\mathbf{x}, \mathbf{x}' \in \mathcal{X}^n$ of datasets that differ only in the data of a single user and all $\alpha > 1$,*

$$D_\alpha \left(M(\mathbf{x}) \parallel M(\mathbf{x}') \right) \leq \alpha\rho. \tag{3}$$

ZCDP can be easily converted to DP:

▶ **Lemma 19** ([13]). *For any $\rho > 0, \delta \in (0, 1/2)$, any ρ-zCDP algorithm is $(\rho + 2\sqrt{\rho \cdot \ln(1/\delta)}, \delta)$-DP.*

It also has a simple composition theorem:

▶ **Lemma 20** ([13]). *Let M be a mechanism that just runs subroutines that are ρ_1-zCDP, ..., ρ_m-zCDP. Then, M is $(\rho_1 + \cdots + \rho_m)$-zCDP.*

Trust Graph zCDP is simply zCDP on the view of the non-neighbors:

▶ **Definition 21** (Trust Graph zCDP). *Let $G = (V, E)$. A protocol P satisfies* (ρ, G)-Trust Graph zCDP ((ρ, G)-TGzCDP) *if for each vertex $v \in V$, $\textsc{view}_P^{V\setminus N[v]}(\mathbf{x})$ satisfies ρ-zCDP with respect to the input x_v for all values of $\mathbf{x}_{-v}$. I.e., for all pairs $x_v, x'_v \in \mathcal{X}$, all values of $\mathbf{x}_{-v}$ and all $\alpha > 1$,*

$$D_\alpha \left(\textsc{view}_P^{V\setminus N[v]}(x_v, \mathbf{x}_{-v}) \;\middle\|\; \textsc{view}_P^{V\setminus N[v]}(x'_v, \mathbf{x}_{-v}) \right) \leq \alpha\rho.$$

We can now state the algorithm for vector summation, which is similar to the algorithm in Theorem 8.

▶ **Theorem 22.** *There is an (ρ, G)-TGzCDP mechanism for the aggregation problem with ℓ_2^2-error $2d\Delta^2|T|/\rho$, where T is any dominating set of G. Furthermore, the estimate is unbiased.*

Proof. Let $\sigma = \Delta\sqrt{\frac{1}{2\rho}}$. The protocol works as follows:

- First, each user $v \in V$ picks an arbitrary vertex $u_v \in T \cap N[v]$.
- Each user $u \in T$ broadcasts the sum of all vectors it receives together with a noise drawn from $\mathcal{N}(0, \sigma^2 I_d)$. More formally, the user broadcasts $a_u = \sum_{\substack{v \in V \\ u_v = u}} x_v + z_u$, where $z_u \sim \mathcal{N}(0, \sigma^2 I_d)$.
- Finally, the estimate is $\tilde{a} = \sum_{u \in T} a_u$.

Privacy Analysis Consider any $v \in V$ and $\mathbf{x}_{-v} \in \mathcal{X}^{V \setminus \{v\}}$. Similar to the proof of Theorem 8, the view is a post-processing of $z_u + x$. Thus, for any $x_v, x'_v \in \mathcal{X}$, we have

$$D_\alpha\left(\text{VIEW}_P^{V \setminus N[v]}(x_v) \,\middle\|\, \text{VIEW}_P^{V \setminus N[v]}(x'_v)\right) \le D_\alpha(z_{u_v} + x_v \parallel z_{u_v} + x'_v) \le \alpha \cdot \rho,$$

where the second inequality follows from the zCDP guarantee of the Gaussian mechanism (e.g., Proposition 16 of [13]). Thus, the protocol satisfies (ρ, G)-TGzCDP as desired.

Utility Analysis It is clear that the estimate is unbiased. The ℓ_2^2-error is

$$\mathbb{E}\left[(\tilde{a} - a)^2\right] = \sum_{u \in T} \mathbb{E}[\|z_u\|^2] = |T| \cdot (\sigma^2 d) = 2d\Delta^2 |T|/\rho. \quad \blacktriangleleft$$

Using Lemma 19, we immediately get the following corollary. For $|T| = \Theta(1)$, this guarantee (asymptotically) matches the known lower bound in central DP (see, e.g., [7]), while for $|T| = \Theta(n)$, this guarantee nearly matches the known lower bound in local DP [4].

▶ **Corollary 23.** *For any $\varepsilon < O(\log(1/\delta))$, there is an (ε, δ, G)-TGDP mechanism for the aggregation problem with ℓ_2^2-error $O\left(\frac{d\Delta^2 |T| \log(1/\delta)}{\varepsilon^2}\right)$, where T is any dominating set of G. Furthermore, the estimate is unbiased.*

5.2 From Vector Summation to Convex Optimizaiton

Using the above vector summation protocol, we can immediately implement several DP ML algorithms in the literature, such as the DP-SGD algorithm [1]. We can also obtain formal guarantee for convex optimization problems from these algorithms. For instance, let us consider the convex ERM problem [7]. Here there is a loss function $\ell : \mathcal{W} \times \mathcal{X} \to \mathbb{R}$ which is L-Lipschitz on the first parameter and suppose that the diameter of $\mathcal{W}$ is at most R. The goal is to minimize the empirical loss $\mathcal{L}(w, \mathbf{x}) := \frac{1}{n} \sum_{v \in V} \ell(w, x_v)$. Using the above vector summation algorithm, we can arrive at the following:

▶ **Theorem 24.** *For $0 < \rho \le O(1)$, there is an (ρ, G)-TGzCDP mechanism for the convex ERM problem with expected excess risk $O\left(\frac{RL\sqrt{d|T|/\rho}}{n}\right)$, where T is any dominating set of G.*

Proof Sketch. The algorithm uses the (stochastic) mirror descent (see, e.g., [12, Section 6.1]) over n^2 steps. In each step, the gradient is computed by running our (ρ', G)-TGzCDP vector summation protocol with $\rho' = \rho/n^2, \Delta = G$ and scale the answer by a factor of $\frac{1}{n}$. The composition theorem (Lemma 20) implies that the algorithm is ρ-TGzCDP. The excess risk guarantee follows from standard analysis of stochastic mirror descent (e.g., [12, Theorem 6.1]). ◀

Again, using Lemma 19, we immediately get the following corollary for (ε, δ)-DP. For $|T| = \Theta(1)$, this guarantee (asymptotically) matches the known lower bound in central DP [7].

▶ **Corollary 25.** *For any $\varepsilon < O(\log(1/\delta))$, there is an (ε, δ, G)-TGDP mechanism for the convex ERM problem with expected excess risk $O\left(\frac{RL\sqrt{d|T|\log(1/\delta)}}{\varepsilon n}\right)$, where T is any dominating set of G.*

6 Conclusion and Future Directions

We have proposed a new model of privacy given a graph of trust relationships between users. Our model generalizes central and local DP, and further captures intermediate trust structures such as social networks or multiple curators. A significant open theoretical problem is to close the gap between the upper and lower bounds for TGDP, though our experiments suggest that this gap may be small in practice.

References

1 Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *CCS*, pages 308–318, 2016. doi:10.1145/2976749.2978318.

2 John M Abowd. The US Census Bureau adopts differential privacy. In *KDD*, pages 2867–2867, 2018.

3 Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018. doi:10.1080/23307706.2017.1397554.

4 Hilal Asi, Vitaly Feldman, and Kunal Talwar. Optimal algorithms for mean estimation under local differential privacy. In *ICML*, pages 1046–1056, 2022. URL: https://proceedings.mlr.press/v162/asi22b.html.

5 Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. The privacy blanket of the shuffle model. In *CRYPTO*, pages 638–667, 2019. doi:10.1007/978-3-030-26951-7_22.

6 Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. Private summation in the multi-message shuffle model. In *CCS*, pages 657–676, 2020. doi:10.1145/3372297.3417242.

7 Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *FOCS*, pages 464–473, 2014. doi:10.1109/FOCS.2014.56.

8 Amos Beimel, Kobbi Nissim, and Eran Omri. Distributed private data analysis: Simultaneously solving how and what. In *CRYPTO*, pages 451–468, 2008. doi:10.1007/978-3-540-85174-5_25.

9 James Henry Bell, Kallista A Bonawitz, Adrià Gascón, Tancrède Lepoint, and Mariana Raykova. Secure single-server aggregation with (poly) logarithmic overhead. In *CCS*, pages 1253–1269, 2020. doi:10.1145/3372297.3417885.

10 Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnés, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *SOSP*, pages 441–459, 2017. doi:10.1145/3132747.3132769.

11 Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *CCS*, pages 1175–1191, 2017.

12 Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3-4):231–357, 2015. doi:10.1561/2200000050.

13 Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *TCC*, pages 635–658, 2016. doi:10.1007/978-3-662-53641-4_24.

14 Alewyn P. Burger, Michael A. Henning, and Jan H. van Vuuren. On the ratios between packing and domination parameters of a graph. *Discrete Mathematics*, 309:2473–2478, 2009. doi:10.1016/J.DISC.2008.05.030.

15 T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Optimal lower bound for differentially private multi-party aggregation. In *ESA*, pages 277–288, 2012. doi:10.1007/978-3-642-33090-2_25.

16 Albert Cheu, Adam D. Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *EUROCRYPT*, pages 375–403, 2019. doi:10.1007/978-3-030-17653-2_13.

17 Albert Cheu and Chao Yan. Necessary conditions in multi-server differential privacy. In *ITCS*, 2023.

18 Edwige Cyffers and Aurélien Bellet. Privacy amplification by decentralization. In *AISTATS*, pages 5334–5353, 2022. URL: https://proceedings.mlr.press/v151/cyffers22a.html.

19 Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *JMLR*, 17(83):1–5, 2016. URL: https://jmlr.org/papers/v17/15-408.html.

20 Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. *NIPS*, 30, 2017.

21 Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503, 2006. doi:10.1007/11761679_29.

22 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006. doi:10.1007/11681878_14.

23 Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014. doi:10.1561/0400000042.

24 Cynthia Dwork and Guy N. Rothblum. Concentrated differential privacy. *CoRR*, abs/1603.01887, 2016. arXiv:1603.01887.

25 Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *SODA*, pages 2468–2479, 2019. doi:10.1137/1.9781611975482.151.

26 Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *PODS*, pages 211–222, 2003. doi:10.1145/773153.773174.

27 Uriel Feige. A threshold of ln n for approximating set cover. *JACM*, 45(4):634–652, 1998. doi:10.1145/285055.285059.

28 Vitaly Feldman. Dealing with range anxiety in mean estimation via statistical queries. In *ALT*, pages 629–640, 2017. URL: http://proceedings.mlr.press/v76/feldman17b.html.

29 Vitaly Feldman, Cristóbal Guzmán, and Santosh S. Vempala. Statistical query algorithms for mean vector estimation and stochastic convex optimization. In *SODA*, pages 1265–1277, 2017. doi:10.1137/1.9781611974782.82.

30 Badih Ghazi, Ravi Kumar, Pasin Manurangsi, and Rasmus Pagh. Private counting from anonymous messages: Near-optimal accuracy with vanishing communication overhead. In *ICML*, pages 3505–3514, 2020. URL: http://proceedings.mlr.press/v119/ghazi20a.html.

31 Badih Ghazi, Ravi Kumar, Pasin Manurangsi, Rasmus Pagh, and Amer Sinha. Differentially private aggregation in the shuffle model: Almost central accuracy in almost a single message. In *ICML*, pages 3692–3701, 2021. URL: http://proceedings.mlr.press/v139/ghazi21a.html.

32 Badih Ghazi, Pasin Manurangsi, Rasmus Pagh, and Ameya Velingker. Private aggregation from fewer anonymous messages. In *EUROCRYPT*, pages 798–827, 2020. doi:10.1007/978-3-030-45724-2_27.

33 Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. *SICOMP*, 41(6):1673–1693, 2012. doi:10.1137/09076828X.

34 Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game, or a completeness theorem for protocols with honest majority. In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pages 307–328. Association for Computing Machinery, 2019. doi:10.1145/3335741.3335755.

35 Magnús M. Halldórsson, Jan Kratochvíl, and Jan Arne Telle. Independent sets with domination constraints. *Discret. Appl. Math.*, 99(1-3):39–54, 2000. doi:10.1016/S0166-218X(99)00124-9.

36 Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography from anonymity. In *FOCS*, pages 239–248, 2006. doi:10.1109/FOCS.2006.25.

37 Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SICOMP*, 40(3):793–826, 2011. doi:10.1137/090756090.

38 Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *JACM*, 45(6):983–1006, 1998. doi:10.1145/293347.293351.

39 Bryan Klimt and Yiming Yang. The Enron corpus: A new dataset for email classification research. In *ECML*, pages 217–226, 2004. doi:10.1007/978-3-540-30115-8_22.

40 Srijan Kumar, Bryan Hooi, Disha Makhija, Mohit Kumar, Christos Faloutsos, and VS Subrahmanian. Rev2: Fraudulent user prediction in rating platforms. In *WSDM*, pages 333–341, 2018.

41 Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. Edge weight prediction in weighted signed networks. In *ICDM*, pages 221–230, 2016.

42 Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *TKDD*, 1(1), 2007. doi:10.1145/1217299.1217301.

43 Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

44 Jure Leskovec and Julian Mcauley. Learning to discover social circles in ego networks. *NIPS*, 25, 2012.

45 László Lovász. On the ratio of optimal integral and fractional covers. *Discret. Math.*, 13(4):383–390, 1975. doi:10.1016/0012-365X(75)90058-8.

46 Carey Radebaugh and Ulfar Erlingsson. Introducing TensorFlow Privacy: Learning with Differential Privacy for Training Data, March 2019. URL: blog.tensorflow.org.

47 Matthew Richardson, Rakesh Agrawal, and Pedro Domingos. Trust management for the semantic web. In *ISWC*, pages 351–368, 2003. doi:10.1007/978-3-540-39718-2_23.

48 Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *CoRR*, abs/1909.13021, 2019. arXiv:1909.13021.

49 Daniel J Solove. Conceptualizing privacy. *Calif. L. Rev.*, 90:1087, 2002.

50 Thomas Steinke. Multi-central differential privacy. *CoRR*, abs/2009.05401, 2020. arXiv:2009.05401.

51 Davide Testuggine and Ilya Mironov. PyTorch Differential Privacy Series Part 1: DP-SGD Algorithm Explained, August 2020. URL: medium.com.

52 Salil Vadhan. The complexity of differential privacy. *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, pages 347–450, 2017. doi:10.1007/978-3-319-57048-8_7.

53 Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *JASA*, 60(309):63–69, 1965.

54 David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.

55 Andrew C-C Yao. Protocols for secure computations. In *FOCS*, pages 160–164, 1982.

56 Andrew C-C Yao. How to generate and exchange secrets. In *FOCS*, pages 162–167, 1986.

A Example Graph

Figure 2 gives an example of a graph with a significant gap between the domination number and the packing number, which importantly illustrates the gap between the upper and lower bounds on MSE under TGDP in Theorems 8, 9, and 10.

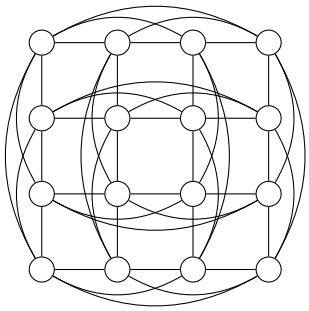


Figure 2 A graph with a gap between the domination number (4) and the packing number (1). The relaxed LP solution $\mathrm{OPT}_{\mathrm{LP}} = 16/7 \approx 2.285$.

B Missing Proofs

Proof of Lemma 6. We may write $Z = Z^1 + Z^2$ where $Z^1 \sim \mathrm{sNB}\left(1, 1 - e^{-\varepsilon/\Delta}\right) = \mathrm{DLap}(\Delta/\varepsilon)$ and $Z^2 \sim \mathrm{sNB}\left(r-1, 1 - e^{-\varepsilon/\Delta}\right)$ are independent. We again can think of $x + Z$ and $x' + Z$ as a post-processing of $x + Z^1$ and $x' + Z^1$, respectively. This implies that

$$\begin{aligned}
&D_\infty\left(x + Z \parallel x + Z\right) \\
&\le D_\infty\left(x + Z_1 \parallel x' + Z_1\right) \\
&= \max_{o \in \mathbb{N}} \ln\left(\frac{\Pr_{Z_1 \sim \mathrm{DLap}(\Delta/\varepsilon)}[x + Z_1 = o]}{\Pr_{Z_1 \sim \mathrm{DLap}(\Delta/\varepsilon)}[x' + Z_1 = o]}\right) \\
&= \frac{\varepsilon}{\Delta} \cdot \left(|o - x| - |o - x'|\right) \\
&\le \frac{\varepsilon}{\Delta} \cdot |x' - x| \le \varepsilon,
\end{aligned}$$

where the last two inequalities follow from the triangle inequality and since $0 \le x, x' \le \Delta$ respectively. ◀

Proof of Theorem 15. Let $\mathbf{y} = (y_u)_{u \in V}$ denote any solution to LP in (2). We use exactly the same protocol as in Theorem 9. The utility analysis is also similar to before. Thus, we only provide the privacy analysis below.

Privacy Analysis

Consider any $v \in V, T \in \binom{N(v)}{\le t_v}$ and $\mathbf{x}_{-v} \in \{0, \dots, \Delta\}^{V \setminus \{v\}}$. We write $\mathbf{a}$ as a shorthand for $(a_u)_{u \in V}$. Notice that $\mathrm{VIEW}_P^{V \setminus (N[v] \setminus T)}(x)$ is exactly $(\mathbf{x}_{-(N[v] \setminus T)}, \mathbf{a}(x), (s^u_{v'})_{u \in V \setminus (N[v] \setminus T), v' \in V})$. We claim that this is a post-processing of $(z_u + s^u_v)_{u \in (N[v] \setminus T)} \cup (s^u_v)_{u \in T}$. This is simply because $\mathbf{x}_{-(N[v] \setminus T)}$, $(a_u)_{u \in V \setminus N[v]}$,

$(s^u_{v'})_{u\in V\setminus N[v],v'\in V}, (s^u_{v'})_{u\in T,v'\in V\setminus\{v\}}$ do not depend on $x_v = x$ at all and are independent of $(z_u + s^u_v)_{u\in(N[v]\setminus T)} \cup (s^u_v)_{u\in T}$; finally, note that $(a_u(x))_{u\in N[v]}$ is a post-processing of $(z_u + s^u_v)_{u\in N[v]}$ since $a_u(x) = (z_u + s^u_v) + \sum_{\substack{v'\in N[V]\setminus\{v\}\\ u_{v'}=u}} s^u_{v'}$.

Now, since $(s^u_v(x))_{u\in N[v]}$ are random elements of $\mathbb{Z}_q$ that sums to x, we also have that $(z_u + s^u_v)_{u\in(N[v]\setminus T)} \cup (s^u_v)_{u\in T}$ are random elements of $\mathbb{Z}_q$ that sums to $x + \sum_{u\in N[v]} z_u$. In other words, $(z_u + s^u_v)_{u\in(N[v]\setminus T)} \cup (s^u_v)_{u\in T}$ is a post-processing of $x + \sum_{u\in(N[v]\setminus T)} z_u$. The remainder of the privacy proof then proceed similarly to that in Theorem 9, where we now use the fact that $\sum_{u\in(N[v]\setminus T)} z_u$ is distributed as $\mathrm{sNB}\left(\sum_{u\in(N[v]\setminus T)} y_u, 1 - e^{-\varepsilon/\Delta}\right)$ and $\sum_{u\in(N[v]\setminus T)} y_u \geq 1$ from (2). ◀

Theorem 16 is an immediate consequence of the following reduction to the LDP model, similar to Theorem 10 and the corresponding reduction (Lemma 11).

▶ **Lemma 26.** *Suppose that there is an $(\varepsilon, G, \mathbf{t})$-RTGDP protocol for integer aggregation. Then, there exists an ε-local DP protocol for integer aggregation with the same MSE for $\rho(G)^{\mathbf{t}}$ users, where $\rho(G)^{\mathbf{t}}$ denotes the size of the largest $(2, \mathbf{t})$-robust packing G.*

Proof of Lemma 26. Let $U = \{u_1, \ldots, u_m\} \subseteq V$ and $(T_v)_{v\in U}$ be the largest $(2, \mathbf{t})$-robust packing in G where $m = \rho(G)$. To avoid ambiguity, we use $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_m)$ to denote the input to the local DP protocol.

To construct the LDP protocol, let $Q_1, \ldots, Q_m \subseteq V$ be any partition of V such that $(N[u_i]\setminus T_{u_i}) \subseteq Q_i$ for all $i \in [m]$. Such a partition exists because $N[u_1]\setminus T_{u_1}, \ldots, N[u_m]\setminus T_{u_m}$ are disjoint by definition of a packing. Let P be any $(\varepsilon, G, \mathbf{t})$-RTGDP protocol for integer aggregation. Our LDP protocol $\tilde{P}$ works simply by running the protocol P where each user $i \in [m]$ assumes the role of all users in Q_i where the input is defined as

$$x_u = \begin{cases} x_i & \text{if } u = u_i \\ 0 & \text{otherwise.} \end{cases}$$

for all $u \in Q_i$. We then output the estimate as produced by P. The MSE of $\tilde{P}$ is obviously the same as that of P.

To see that this satisfies ε-LDP, consider any $i \in [m], \tilde{\mathbf{x}}_{-i} \in \mathcal{X}^{[m]\setminus\{i\}}$, we have $\textsc{view}^{[m]\setminus\{i\}}_{\tilde{P}}(\tilde{x}) = \textsc{view}^{V\setminus Q_i}_{P}(\mathbf{x}(\tilde{x}))$ where $\mathbf{x}(\tilde{x})$ is the input to P as defined above. Since $V\setminus Q_i \subseteq V\setminus(N[u_i]\setminus T_{u_i})$, $\textsc{view}^{V\setminus Q_i}_{P}(\mathbf{x}(\tilde{x}))$ is a post-processing of $\textsc{view}^{V\setminus(N[u_i]\setminus T_{u_i})}_{P}(\mathbf{x}(\tilde{x}))$. Thus, for every $\tilde{x}_i, \tilde{x}'_i \in \mathcal{X}$, Lemma 5 implies that

$$\begin{aligned} &D_\infty\left(\textsc{view}^{[m]\setminus\{i\}}_{\tilde{P}}(\tilde{x}_i) \parallel \textsc{view}^{[m]\setminus\{i\}}_{\tilde{P}}(\tilde{x}'_i)\right) \\ &\leq D_\infty\left(\textsc{view}^{V\setminus(N[u_i]\setminus T_{u_i})}_{P}(\tilde{x}_i) \parallel \textsc{view}^{V\setminus(N[u_i]\setminus T_{u_i})}_{P}(\tilde{x}'_i)\right) \\ &\leq \varepsilon, \end{aligned}$$

where the last inequality is due to P being $(\varepsilon, G, \mathbf{t})$-RTGDP protocol.

As a result, $\tilde{P}$ is ε-LDP as desired. ◀

Proof of Theorem 17. We will write $\mathbf{t}'$ as a shorthand for $\mathbf{t} + \lceil\alpha \cdot \deg_U\rceil$.

We claim that $\mathrm{OPT}^{\mathbf{t}'}_{\mathrm{pack}} \geq \gamma \cdot \mathrm{OPT}^{\mathbf{t}}_{\mathrm{LP}}$ for $\gamma = \frac{\alpha}{8}$. To prove this, first observe that LP duality implies that $\mathrm{OPT}^{\mathbf{t}}_{\mathrm{LP}}$ is also equal to the optimal of the following LP:

$$\max \sum_{v\in V, T\in\binom{N(v)}{\leq t_v}} w_{v,T} \tag{4}$$

$$\text{s.t.} \sum_{\substack{u\in V, T\in\binom{N(u)}{\le t_u} \\ (N[u]\setminus T)\ni v}} w_{u,T} \le 1 \qquad \forall v \in V \tag{5}$$

$$0 \le w_{v,T} \le 1 \qquad \forall v \in V, T \in \binom{N(v)}{\le t_v}.$$

For $T' \subseteq N[v']$, we let $\overline{T}'[v'] := N[v'] \setminus T'$.

Given an optimal solution $(w_{v,T})_{v\in V, T\in\binom{N(v)}{\le t_v}}$ to the dual LP (4), we construct a $(2,\mathbf{t}')$-robust packing as follows:

- Include each (v,T) in the set R_0 with probability $2\gamma \cdot w_{v,T}$.
- Filter elements in R_0 to create R_1 where R_1 only includes (v,T) such that

$$v \notin \bigcup_{(v',T')\in(R_0\setminus\{(v,T)\})} \overline{T}'[v'], \tag{6}$$

and

$$\left|\bigcup_{(v',T')\in(R_0\setminus\{(v,T)\})} (N(v)\cap\overline{T}'[v'])\right| \le \alpha\cdot\deg(v). \tag{7}$$

- Construct R_2 by adding $\left(v, T\cup\bigcup_{(v',T')\in R_1\setminus\{(v,T)\}}(N(v')\cap\overline{T}'[v'])\right)$, for each $(v,T)\in R_1$.

By the conditions imposed when constructing R_1, it is simple to see that R_2 is a valid $(2,\mathbf{t})$-robust packing. Thus, we are only left to argue that it has a large size. To do so, first observe that

$$\begin{aligned}
&\mathbb{E}[\,|R_2|\,] \\
&= \mathbb{E}[\,|R_1|\,] \\
&= \sum_{v\in V, T\in\binom{N(v)}{\le t_v}} \Pr[(v,T)\in R_1] \\
&= \sum_{v\in V, T\in\binom{N(v)}{\le t_v}} \Pr[(v,T)\in R_0]\cdot\Pr[(v,T)\in R_1 \mid (v,T)\in R_0] \\
&= \sum_{v\in V, T\in\binom{N(v)}{\le t_v}} (2\gamma\cdot w_{v,T})\cdot\Pr[(v,T)\in R_1 \mid (v,T)\in R_0],
\end{aligned} \tag{8}$$

where the last equality is due to the randomized rounding in the first step.

Let us now fix $v\in V$ and $T\in\binom{N(v)}{\le t_v}$. To bound $\Pr[(v,T)\in R_1 \mid (v,T)\in R_0]$, note that from our procedure, we have

$$\begin{aligned}
&\Pr[(v,T)\in R_1 \mid (v,T)\in R_0] \\
&= \Pr\left[(6) \text{ and } (7) \text{ both hold}\right] \\
&\ge 1 - \Pr[(6) \text{ fails}] - \Pr[(7) \text{ fails}].
\end{aligned} \tag{9}$$

We bound each of the two terms separately. For the first term, we have

$$\begin{aligned}
&\Pr[(6) \text{ fails}] \\
&= \Pr\left[v \in \bigcup_{(v',T')\in(R^0\setminus\{(v,T)\})} \overline{T}'[v']\right]
\end{aligned}$$

$$
\begin{aligned}
&= \Pr\left[\exists \begin{smallmatrix} v' \in V, T' \in \binom{N(v')}{\leq t_{v'}} \\ (N[v'] \setminus T') \ni v \end{smallmatrix}, (v', T') \in R^0\right] \\
&\leq \sum_{\substack{v' \in V, T' \in \binom{N(v')}{\leq t_{v'}} \\ (N[v'] \setminus T') \ni v}} \Pr[(v', T') \in R^0] \\
&= \sum_{\substack{v' \in V, T' \in \binom{N(v')}{\leq t_{v'}} \\ (N[v'] \setminus T') \ni v}} 2\gamma \cdot w_{v',T'} \\
&\overset{(5)}{\leq} 2\gamma \\
&\leq \frac{1}{4}.
\end{aligned} \tag{10}
$$

Similarly, we have

$$
\begin{aligned}
&\Pr[(7) \text{ fails}] \\
&= \Pr\left[\left|\bigcup_{(v',T') \in (R^0 \setminus \{(v,T)\})} (N(v) \cap \overline{T}'[v'])\right| > \alpha \cdot \deg(v)\right] \\
&\leq \frac{\mathbb{E}\left[\left|\bigcup_{(v',T') \in (R^0 \setminus \{(v,T)\})} (N(v) \cap \overline{T}'[v'])\right|\right]}{\alpha \cdot \deg(v)} \\
&= \frac{\sum_{u \in N(v)} \Pr\left[u \in \bigcup_{(v',T') \in (R^0 \setminus \{(v,T)\})} \overline{T}'[v']\right]}{\alpha \cdot \deg(v)} \\
&= \frac{\sum_{u \in N(v)} \Pr\left[\exists \begin{smallmatrix} v' \in V, T' \in \binom{N(v')}{\leq t_{v'}} \\ (N[v'] \setminus T') \ni u \end{smallmatrix}, (v', T') \in R^0\right]}{\alpha \cdot \deg(v)} \\
&\leq \frac{\sum_{u \in N(v)} \sum_{\substack{v' \in V, T' \in \binom{N(v')}{\leq t_{v'}} \\ (N[v'] \setminus T') \ni u}} \Pr[(v', T') \in R^0]}{\alpha \cdot \deg(v)} \\
&= \frac{\sum_{u \in N(v)} \sum_{\substack{v' \in V, T' \in \binom{N(v')}{\leq t_{v'}} \\ (N[v'] \setminus T') \ni u}} 2\gamma \cdot w_{v',T'}}{\alpha \cdot \deg(v)} \\
&\overset{(5)}{\leq} \frac{\sum_{u \in N(v)} 2\gamma}{\alpha \cdot \deg(v)} \\
&= \frac{2\gamma}{\alpha} \leq \frac{1}{4},
\end{aligned} \tag{11}
$$

where the first inequality is due to Markov and the last inequality is from our choice of γ.

Combining Equations (8)–(11), we get

$$
\mathbb{E}[\,|R_2|\,] \geq \sum_{v \in V, T \in \binom{N(v)}{\leq t_v}} \gamma \cdot w_{v,T} = \gamma \cdot \mathrm{OPT}^{\mathbf{t}}_{\mathrm{LP}},
$$

which concludes the proof. ◀

C Additional Experiment Details

We next provide additional experiment details. Optimization problems were solved using `cvxpy` [19, 3].

C.1 Additional Dataset Details

Table 1 lists additional details for each dataset. All datasets are publicly available through SNAP [43]. We give additional dataset descriptions below.

Table 1 Integrality gap comparison of $\mathrm{OPT}_{\mathrm{LP}}$ to minimum dominating set size $|T|$ for 9 network datasets.

Dataset	Number of nodes n	Number of edges $\lvert E\rvert$	Maximum degree
EU Emails (Core)	1,005	16,706	347
Bitcoin (Alpha)	3,783	12,972	507
Facebook	4,039	88,234	1,045
Bitcoin (OTC)	5,881	18,591	788
Enron Emails	36,692	183,831	1,383
GitHub	37,700	289,003	9,458
Epinions	75,879	405,740	3,044
Twitter	81,306	1,342,310	3,383
EU Emails (All)	265,214	365,570	7,636

EU Emails datasets. This data comes from an email network at a European research institution collected by [42]. We include an undirected edge between sender and receiver who have exchanged an email, though the original dataset contains directed edge information. We consider two subgraphs: *(i)* a "core" subgraph consisting of email addresses within the research institution (which we refer to as *EU Emails (Core)*), and *(ii)* the full graph of all emails contained in the dataset (which we refer to as *EU Emails (All)*).

Enron Emails. This data comes from an email communication network within Enron [39]. The graph contains an undirected edge between a sender and receiver if at least one email was exchanged between them. The original graph is undirected.

Bitcoin datasets. We include network data from two different Bitcoin trading platforms, *Bitcoin OTC* and *Bitcoin Alpha* [41, 40]. In the original data, each user rater another user with a value between -10 and 10, where a negative rating corresponds to mistrust and a positive rating corresponds to trust. In our analysis, we include an undirected edge between users if and only if one user rates another with a value greater than 0. We ignore mistrust ratings. In general, further analysis that includes the mistrust ratings would be interesting to conduct.

Facebook dataset. This data from Facebook was published by [44]. Each undirected edge indicates a friend relationship between two users.

GitHub dataset. This data comes from a social network of GitHub developers collected by [48]. Edges represent mutual follower relationships between two users.

Epinions dataset. This data comes from the consumer review site Epinions.com [47]. We include an undirected edge if one user "trusts" another by giving a positive rating to the other user (which indicates trusting their reviews). The original dataset is a directed graph.

Twitter dataset. This data encodes follower relationships from Twitter [44]. We include an undirected edge if one user has follows another. The original dataset contains directed edges for follower relationships.

C.2 Integrality Gap Between Linear Program Optimum and Minimum Dominating Set

The proposed LP-based mechanism for achieving Trust Graph DP presented in Theorem 9 will always perform at least as well as the dominating set protocol in terms of error. Here we consider whether in practice, the LP-based mechanism is better than the dominating

set protocol. We observed an integrality gap between $\mathrm{OPT}_{\mathrm{LP}}$ and the size of the minimum dominating set $|T|$ on three out of nine datasets. Notably, we only observed the integrality gap on the email communication datasets, and not on the Bitcoin or social network datasets. Theoretically, it is known that the integrality gap between $\mathrm{OPT}_{\mathrm{LP}}$ and $|T|$ can be up to a factor of up to $O(\log(n))$ [54]. Further study of the integrality gaps that might arise in other real network settings remains an interesting open question. Table 2 lists the ratio $\frac{\mathrm{OPT}_{\mathrm{LP}}}{|T|}$ for each dataset.

Whether any given graph exhibits an integrality gap or not, a prevailing advantage of the proposed improved algorithm via linear programming for TGDP over a minimum dominating set protocol lies in computational efficiency, as finding the minimum dominating set is NP-hard.

Table 2 Integrality gap comparison of $\mathrm{OPT}_{\mathrm{LP}}$ to minimum dominating set size $|T|$ for 9 network datasets.

Dataset	n	$\mathrm{OPT}_{\mathrm{LP}}$	$\lvert T\rvert$	$\frac{\mathrm{OPT}_{\mathrm{LP}}}{\lvert T\rvert}$
EU Emails (Core)	1,005	111.97	128	0.8748
Bitcoin (Alpha)	3,783	686	686	1
Facebook	4,039	10	10	1
Bitcoin (OTC)	5,881	1,126	1,126	1
Enron Emails	36,692	3,060.66	3,062	0.9996
GitHub	37,700	4,538	4,538	1
Epinions	75,879	15,734	15,734	1
Twitter	81,306	961	961	1
EU Emails (All)	265,214	18,074.40	18,181	0.9941

C.3 Other Local DP Mechanisms for Aggregation

Note that we focused our comparisons to the LDP version of the Laplace mechanism since the ratio has a simple expression. Nevertheless, we point out that there are other LDP mechanisms for aggregation. For example, one can apply randomized rounding to the input (where we set it to Δ with probability x_i/Δ and set it to zero otherwise) before applying the classic randomized response (RR) mechanism [53]. The MSE of this method is actually input-dependent due to the randomized rounding, making it harder to compare against our method. To give the benefit to this algorithm, let us assume for the moment that all inputs are either 0 or Δ. In this case, there is no error due to the randomized rounding. A simple calculation shows that the error from here is $c_\epsilon \cdot 2\Delta^2 n/\epsilon^2$ where $c_\epsilon = \frac{e^\epsilon \cdot \epsilon^2}{2(e^\epsilon - 1)^2}$. For $\epsilon \leq 1$, this constant c_ϵ is at least 0.46. Thus, in all cases, our mechanism still demonstrates a significant improvement over this over-optimistic estimate of the error for randomized rounding + RR.

D Relating Packing Number and Minimum Dominating Set

In this section, we prove Theorem 12. Our proof follows that of Halldorsson et al. [35], who gave a greedy algorithm that provides a $\sqrt{n}$-approximation for the packing number. At a high level, the main difference between our proof and theirs is that we compare the greedy solution with the (optimal) LP solution, whereas they compare it with the (optimal) integral solution. The LP for the packing number turns out to be exactly the dual of the LP for the minimum dominating set (i.e., LP (1)); this then gives us the desired claim.

Proof of Theorem 12. Consider the dual of LP (1), which can be written as follows:

$$\min \sum_{u \in V} y_u \tag{12}$$

$$\text{s.t.} \sum_{u \in N[v]} y_u \leq 1 \qquad \forall v \in V \tag{13}$$

$$0 \leq y_u \leq 1 \qquad \forall u \in V.$$

Let $\mathbf{y}^* = (y^*_u)_{u \in V}$ denote an optimal solution of the LP (12). Consider the following greedy algorithm by Halldorsson et al. [35].

- Set $S \leftarrow \emptyset, i \leftarrow 0$, and $V_i \leftarrow V$.
- While $V_i \neq \emptyset$ do:
  - Let v_i be the vertex in V_i with the smallest degree (ties broken arbitrarily).
  - Set $S \leftarrow S \cup \{v_i\}$.
  - Let $Z_i := \{u \in V_i \mid N[u] \cap N[v_i] \neq \emptyset\}$.
  - Set $V_{i+1} \leftarrow V_i \setminus Z_i$.
  - Set $i \leftarrow i + 1$.
- Output S.

It is clear that the output S is indeed a packing; let $q = |S|$. We claim the following for all $i \in [q]$:

$$\sum_{u \in Z_i} y^*_u \leq \sqrt{n}. \tag{14}$$

Before we prove (14), let us argue that it implies $\rho(G) \geq \mathrm{OPT}_{\mathrm{LP}}/\sqrt{n}$. Notice that $\{Z_i\}_{i \in [q]}$ is a partition of V. Therefore, we have

$$\mathrm{OPT}_{\mathrm{LP}} = \sum_{u \in V} y^*_u = \sum_{i \in [q]} \sum_{u \in Z_i} y^*_u \overset{(14)}{\leq} \sum_{i \in [q]} \sqrt{n} \leq \rho(G) \cdot \sqrt{n},$$

as desired.

Finally, we prove (14). Consider two cases based on whether $\deg(v_i) \leq \sqrt{n} - 1$.

- Case I: $\deg(v_i) \leq \sqrt{n} - 1$. In this case, we have

$$\sum_{u \in Z_i} y^*_u \leq \sum_{w \in N[v_i]} \sum_{u \in N[w]} y^*_u \overset{(13)}{\leq} \sum_{w \in N[v_i]} 1 = \deg(v_i) + 1 \leq \sqrt{n}.$$

- Case II: $\deg(v_i) > \sqrt{n} - 1$. Since we pick v_i to be the vertex with the smallest degree among those in V_i, we have $\deg(u) > \sqrt{n} - 1$ for all $u \in Z_i$. Therefore, we have

$$\begin{aligned}
\sum_{u \in Z_i} y^*_u &\leq \frac{1}{\sqrt{n}} \cdot \sum_{u \in Z_i} (\deg(u) + 1) \cdot y^*_u \\
&\leq \frac{1}{\sqrt{n}} \cdot \sum_{u \in V} (\deg(u) + 1) \cdot y^*_u \\
&= \frac{1}{\sqrt{n}} \cdot \sum_{v \in V} \sum_{u \in N[v]} y^*_u \\
&\overset{(13)}{\leq} \frac{1}{\sqrt{n}} \cdot n \\
&= \sqrt{n}.
\end{aligned}$$

Thus, in both cases (14) holds, which completes our proof. ◀

Partial Minimum Branching Program Size Problem Is ETH-Hard

Ludmila Glinskih
Google, Mountain View, CA, USA

Artur Riazanov
EPFL, Lausanne, Switzerland

Abstract

We show that assuming the Exponential Time Hypothesis, the Partial Minimum Branching Program Size Problem (MBPSP*) requires superpolynomial time. This result also applies to the partial minimization problems for many interesting subclasses of branching programs, such as read-k branching programs and OBDDs.

Combining these results with the recent unconditional lower bounds for MCSP [6], we obtain an unconditional superpolynomial lower bound on the size of Read-Once Nondeterministic Branching Programs (1-NBP) computing the *total* versions of the minimum BP, read-k-BP, and OBDD size problems.

Additionally we show that it is **NP**-hard to check whether a given BP computing a partial Boolean function can be compressed to a BP of a given size.

2012 ACM Subject Classification Theory of computation → Circuit complexity

Keywords and phrases MCSP, branching programs, meta-complexity, lower bounds

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.54

Funding Artur Riazanov was supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number MB22.00026. Ludmila Glinskih was supported by a Sloan Research Fellowship to Mark Bun. Most of this work was done while Ludmila was at Boston University, and in part while she was visiting the Simons Institute for the Theory of Computing.

Acknowledgements The authors are grateful to Rahul Santhanam for suggesting working on connections between meta-complexity and branching programs, and to Mark Bun and anonymous reviewers for very helpful feedback on the preliminary versions of this paper.

1 Introduction

The Minimum Circuit Size Problem (MCSP) has been a central problem of study in computational complexity in recent years. Given the truth-table of a Boolean function, and a size parameter s, MCSP asks one to determine whether the function can be computed by a Boolean circuit of size s. As the input length of MCSP is exponential in the input length of the function defined by the given truth table, it is clear that MCSP is contained in **NP**. Although it is not known whether MCSP is **NP**-complete, it is widely believed to be hard to compute. Starting from the work of Kabanets and Cai [11] there has been mounting evidence that MCSP $\notin$ **P** [20]. For example, MCSP $\in$ **P** implies that widely believed cryptographic assumptions do not hold. On the other hand, there are complexity-theoretic barriers to proving **NP**-hardness of MCSP [15, 21].

Even before the work of Kabanets and Cai [11], variations of MCSP were studied in the context where the function is given not as a truth table, but represented in some computational model. One family of models that was extensively studied in this context is *branching programs*. The line of work [2, 17, 23, 22, 1] showed that branching program minimization is **NP**-hard for the natural cases of succinct representation of a function: either

© Ludmila Glinskih and Artur Riazanov;
licensed under Creative Commons License CC-BY 4.0

the function itself is represented as a branching program as in [2, 22, 1], or an input is a *partial function* with few fixed values as in [23]. The results, however, are limited to weak models of branching programs such as OBDDs or read-once branching programs. To our knowledge, no hardness results are known even for minimizing read-twice branching programs.

The potential cross-pollination between these two research areas motivates us to study Minimum Branching Program Size Problem (MBPSP): given a function represented as a truth table $\{0,1\}^{2^n}$ and a size parameter $s \in [2^n]$ determine whether exists a *branching program* of size at most s computing the given function.

On the meta-complexity side, one of the approaches to prove **NP**-hardness of MCSP is to show lower bounds for $\mathcal{C}$-MCSP problems, in which we ask whether an input Boolean function can be computed by a circuit from the class $\mathcal{C}$, where $\mathcal{C}$ is a weaker class than general Boolean circuits. There are already results showing **NP**-hardness of DNF-MCSP [14], (DNF$\circ$XOR)-MCSP [8], and the hardness of DeMorgan Formulas-MCSP (MFSP) assuming the Exponential Time Hypothesis (ETH) [10]. All of these reductions share a common structure.

- First they show hardness of the Partial $\mathcal{C}$-MCSP ($\mathcal{C}$-MCSP*) problem in which the input is the truth table of a partial function.
- Then they show a reduction from the $\mathcal{C}$-MCSP* problem to the $\mathcal{C}$-MCSP problem.

Hardness results for Partial MCSP (MCSP*) are already known. In [9] Ilango showed that MCSP* requires superpolynomial time to compute[1], assuming the Exponential Time Hypothesis. Recently, Hirahara [7] showed the **NP**-hardness of MCSP* under randomized reductions. So the only missing component for showing hardness of MCSP is to show an efficient reduction from MCSP* to MCSP.

The study of hardness of MBPSP under the ETH continues the line of work above. Expressive power of branching programs is situated between that of Boolean formulas and Boolean circuits [18, 5]. On the other hand our understanding of branching programs seems to be slightly better than our understanding of circuits. One evidence for this is that the strongest lower bound on the BP size for an explicit function is $\tilde{\Omega}(n^2)$ [16] which is much stronger than $\Omega(n)$ lower bound for Boolean circuits [3]. Showing the ETH-hardness of MBPSP would improve the recent ETH-hardness result for MFSP [10] which is the frontier hardness result for total circuit minimization problems and move us one step closer to the ETH-hardness of MCSP itself. Our main result is completing the first step of the common recipe for establishing the hardness of $\mathcal{C}$-MCSP by showing hardness of BP-MCSP* (that we call MBPSP* for short).

▶ **Theorem 1.** *Every deterministic algorithm computing MBPSP* on inputs of length N requires time $N^{\Omega(\log\log(N))}$ assuming the Exponential Time Hypothesis.*

Note that the ETH-hardness of MCSP* or MFSP* does not imply the ETH-hardness of MBPSP*. Although on surface it seems that MBPSP (MBPSP*) is just a special case of MCSP (MCSP*), the hardness of the latter does not necessarily imply the hardness of the former. For example, in some special settings of parameters, an explicit construction of a many-one reduction between MCSP and MBPSP yields breakthrough circuit or BP

[1] The same hardness result holds for Partial MFSP (MFSP*).

lower bounds.[2] On the other hand, ruling out more natural parameter configurations does not seem as trivial: indeed, our main result shows that the identity function is a reduction on a certain subset of the instances.

On the branching program minimization side, the consequences are twofold. First, our result implies that all partial branching program minimization problems require time $N^{\Omega(\log\log N)}$ under ETH for any representation of a function that is at least as efficient as truth table.[3] Second, we show that our result implies **NP**-hardness of minimization of a branching program computing a partial function that itself is represented as a branching program (see Section 1.2.3 for details).

In particular, our result implies that the problem of finding the smallest OBDD (a subclass of branching programs) that computes the given partial function is ETH-hard. But surprisingly, there is a polynomial-time algorithm that finds the smallest OBDD computing a *total* function [4]. This means that assuming ETH holds, the potential reduction from MBPSP* to MBPSP must fail on OBDDs.

1.1 Techniques

To show Theorem 1 we adapt the proof of a similar result of Ilango [9] for the hardness of MCSP* and MFSP* problems.

▶ **Theorem 2** ([9]). *Every deterministic algorithm computing MCSP* on inputs of length N requires time $N^{\Omega(\log\log(N))}$ assuming the Exponential Time Hypothesis holds. The same holds for MFSP*.*

Ilango's proof of Theorem 2 is a reduction from the $(n \times n)$-Bipartite Permutation Independent Set Problem ($(n \times n)$-BPIS). For now we can think of it as an independent set problem on some special set of graphs. This is an **NP**-hard graph problem introduced in [13], where the authors show that any deterministic Turing machine requires $2^{\Omega(n \log n)}$-time[4] to solve it if the Exponential Time Hypothesis holds. For an input graph G with vertices labeled by elements of $[n] \times [n]$ Ilango constructs a truth-table of a $3n$-variable partial Boolean function γ_G, such that

($\star$) G is a YES-instance of $(n \times n)$-BPIS iff γ_G can be computed by a DeMorgan formula with at most $3n - 1$ gates.

The key point here is that if we constrain a circuit or a formula to have at most $3n - 1$ (inner) gates, its structure becomes simple enough for a very precise analysis. [9] characterizes the structure of such formulas that compute γ_G which would yield the correspondence between such formulas with independent sets in G.

Although branching programs are situated between formulas and circuits in computational expressivity [5], Theorem 2 does not apply to MBPSP*, since with a sharp constraint on the size, branching programs may become stronger than circuits. We observe in Proposition 7

[2] Suppose that f is such a reduction from $\mathrm{MCSP}_{s=3n}$ (the language of the truth-tables computable by circuits with at most $3n$ gates) to $\mathrm{MBPSP}_{s=n^{2.1}}$. Then taking any function requiring circuits of size larger than $3n$ [3, 12] and applying f to it yields $n^{2.1}$ lower bound for general branching programs superior to the current state of the art. In a similar way, we get superlinear circuit lower bounds from any reduction from $\mathrm{MBPSP}_{s=n^2}$ to $\mathrm{MCSP}_{s=n^{1+\varepsilon}}$.

[3] We note that results in this area often establish **NP**-hardness and even hardness of size approximation for the case of concise representation, while our result only establishes ETH-hardness of the exact version of the problem.

[4] The input to this problem has $\Theta(n^4 \log n)$ bit size.

that this is indeed the case, showing that branching programs are stronger than circuits if we constrain the size parameter of both models to "minimal viable" size: $m-1$ inner nodes for a DeMorgan circuit and m non-sink nodes for a BP for the case of a Boolean function on m input bits.

Nevertheless, we show that the reduction to MCSP* from Theorem 2 does yield ETH-hardness for MBPSP*. Indeed, in our proof of Theorem 1 we only change the part ($\star$), keeping the construction of γ_G. The key part of the proof is to understand what structure a branching program has if it has at most $3n$ (non-sink) nodes. As γ_G is sensitive in every input variable, we need to understand a structure of a branching program that has unique variable labels. We call such programs *once-appearance branching programs* (oaBP). We show that γ_G can be computed by a once-appearance branching program iff G is a YES-instance of $(n \times n)$-BPIS.

1.2 Corollaries

1.2.1 Weaker Classes

BP is not the only model that collapses to oaBP when constrained to have at most n non-sink nodes. The same also holds for OBDD, and k-BP for every positive integer k. From this we get a family of hardness of minimization results for restricted classes of branching programs.

▶ **Corollary 3.** *Every deterministic algorithm computing $\mathcal{C}$-MCSP* for $\mathcal{C} \in \{\text{OBDD}, \text{BP}\} \cup \{k\text{-BP} \mid k \geq 1\}$[5] on inputs of length N requires time $N^{\Omega(\log\log(N))}$ assuming the Exponential Time Hypothesis.*

1.2.2 Unconditional Results for a Weaker Model

In the recent work [6] the authors employed the framework of Ilango from [9] to show an unconditional lower bound on the size of the read-once nondeterministic branching program (1-NBP) computing the total version of MCSP. The proof consists of three components.

1. Show that $(n \times n)$-BPIS unconditionally cannot be computed by a 1-NBP of smaller than $\Omega(n!)$ size.
2. Show that Ilango's reduction from $(n \times n)$-BPIS to MCSP* can be efficiently encoded using BP. This shows that if MCSP* on inputs of length N could be computed by a 1-NBP of size $N^{o(\log\log(N))}$, then $(n \times n)$-BPIS could be computed by a 1-NBP of size $o(n!)$.
3. Show that MCSP and MCSP* have the same 1-NBP complexity up to a linear multiplicative factor. This implies that 1-NBP complexity of MCSP is at least $N^{\Omega(\log\log(N))}$.

Since the proof of Theorem 1 follows similar steps as the proof of the Theorem 2, and analogs of the third property from [6] also hold for BP, k-BP, and OBDD, we get similar unconditional 1-NBP lower bounds for total minimization problems in all these models.

▶ **Corollary 4.** *Every* 1-NBP *computing $\mathcal{C}$-MCSP for $\mathcal{C} \in \{\text{OBDD}, \text{BP}\} \cup \{k\text{-BP} \mid k \geq 1\}$ on inputs of length N requires size at least $N^{\Omega(\log\log(N))}$.*

[5] Or any other class of branching programs that collapses to oaBP on the n-node programs.

1.2.3 NP-hardness of Compressing BPs

Additionally, we show that given a graph G we can construct a branching program B of polynomial size that exactly computes the partial function γ_G. Namely it outputs $\{0,1\}$ values on inputs on which γ_G is defined and $*$ on all other inputs. Then B can be compressed to an oaBP which agrees with γ_G on its domain iff G is a YES-instance of $(n \times n)$-BPIS. Combining this result with a folklore result of **coNP**-hardness of compressing a branching program, we get the following result on the hardness of compressing branching programs.

▶ **Corollary 5.** *The language of pairs (B, s) where B is a branching program[6] over the alphabet $\{0,1,*\}$ and s is in integer, such that an extension of the partial function described by B can be computed by a* BP *of size at most s, is* **NP** *and* **coNP***-hard.*

1.3 Organization

The structure of the paper is as follows:

- In Section 2 we give definitions of the branching programs and various versions of MCSP we are working with.
- In Section 3 we state our main result: the hardness of MBPSP* under ETH (Theorem 1) and explain the main ideas behind its proof. We show the proof of Theorem 1 and proofs of all required lemmas in Section 4 and Section 5.
- In Section 6 we sketch the proofs of several corollaries of Theorem 1.
- Finally, in Section 7 we explain the future directions of this work and possible barriers we see for strengthening our results.

2 Preliminaries

Throughout this work we denote by $[n]$ the set of n elements $\{1, \ldots, n\}$. Let a and b be partial assignments. If the supports of a and b do not intersect, denote by $a \cup b$ the partial assignment that coincides with a on the support of a and with b on the support of b. For a graph G we denote its set of vertices by $V(G)$ and its set of edges by $E(G)$. Throughout the paper a *circuit* is a DeMorgan circuit ($\wedge$, $\vee$ and $\neg$ gates) with gates of arity 2. A *formula* is a tree-like DeMorgan circuit. A *read-once formula* is a formula such that each input bit feeds into exactly one gate. E.g. $x \vee (\neg y \wedge z)$ is a read-once formula, and $(x \wedge y) \vee (\neg x \wedge z)$ is not. Since every DeMorgan formulas can be transformed in such a way that all $\neg$-gates are adjacent to the inputs, we identify DeMorgan formulas with trees with *literals* in the leaves and $\wedge/\vee$ in the inner nodes.

A *Branching Program* (BP) is a form of representation of functions. A n-ary function[7] $f\colon D^n \to \{0,1\}$, where $|D| = d$, is represented by a directed acyclic graph with exactly one source and two sinks. Each non-sink node is labeled with a variable; every internal node has exactly d outgoing edges: labeled with elements of D. One of the sinks is labeled with 1 and the other is labeled with 0. We say that a node *queries* x if it is labeled with a variable x. The *size* of a branching program is the number of nodes in it.

The value of the function for given values of the variables is evaluated as follows: we start in the source and then follow the edge labeled with the value of the variable in the source, we repeat this process until the current node is a sink. The label of this sink is the value of the function. We refer to a path that terminates in a σ-labeled sink as *σ-path*.

[6] The same is true for all sub-classes of branching programs that simulate 4-BP.
[7] We will instantiate this definition for $D = \{0,1\}$ and $D = \{0,1,*\}$

A *Non-deterministic Branching Program* (NBP) is a Branching Program over the alphabet $\{0,1\}$ with additional non-labeled *guessing* nodes with two non-labeled outgoing edges. The value computed by a Non-deterministic Branching Program is 1 if there exists at least one 1-path that agrees with the given assignment.

A Branching Program (deterministic or not) is called *read-once* (denoted as 1-BP in the deterministic case and 1-NBP in the non-deterministic case) if any path from the source to a sink contains each variable label at most once.

For a partial assignment $\alpha\colon [n] \to \{0,1,*\}$ and a branching program B with n input bits over the alphabet $\{0,1\}$ we write $B|_\alpha$ for a branching program obtained by removing every edge of form $x_i \neq \alpha(i)$ and contracting every edge of form $x_i = \alpha(i)$ if $\alpha(i) \in \{0,1\}$. It is easy to see that if B computes the function f, then $B|_\alpha$ computes the function $f|_\alpha$.

The Minimum Branching Program Size Problem (MBPSP(f,s)) gets as input the truth-table of a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ of length $N = 2^n$ and a size parameter s, and outputs 1 if there exists a branching program of size at most s that computes f. We use notation $\mathrm{MBPSP}_{s'(n)}(f) = \mathrm{MBPSP}(f, s'(n))$.

The Partial Minimum Branching Program Size Problem (MBPSP$^*(f^*,s)$) gets as input the partial truth-table of a function $f^*\colon \{0,1\}^n \to \{0,1,*\}$ of length $N = 2^n$ and a size parameter s, and outputs 1 if there exists a substitution of each $*$ in the truth-table to a 0/1, transforming f^* to a Boolean function f, such that $\mathrm{MBPSP}(f,s) = 1$. We use notation $\mathrm{MBPSP}^*_{s'(n)}(f) = \mathrm{MBPSP}^*(f, s'(n))$.

2.1 Once-Appearance Branching Programs

A *once-appearance branching program* (oaBP) is a branching program with distinct node labels. Hence, for a function depending on n variables such oaBP representation contains at most n non-sink nodes. For oaBP we identify the non-sink nodes with their labels, and identify the edges with equalities of form $x = \alpha$ for a variable x and α from the alphabet.

It is easy to see, that oaBPs are strictly stronger than read-once DeMorgan formulas.[8]

▶ **Proposition 6.** *If f is computable by a read-once DeMorgan formula, then it is computable by an* oaBP*.*

Proof. We first observe that if g and h are computable by oaBPs, then so are $g \wedge h$ and $g \vee h$ whenever g and h have disjoint set of variables. Indeed oaBP for $g \wedge h$ is constructed by redirecting all edges going to the 1-in the oaBP computing g to the source of the oaBP computing h, and by redirecting all edges to the 0-sink in g to the 0-sink of h. For $g \vee h$ we redirect all edges to the 0-sink of to the source of the oaBP for h, and all edges to the 1-sink to the 1-sink of h. Now to finish the proof observe that all literals $\{x_1, \neg x_1, \ldots, x_n, \neg x_n\}$ are trivially computable by an oaBP. ◀

▶ **Proposition 7.** *Let $g(x,y,z) \coloneqq (y \wedge x) \vee (z \wedge \neg x)$. Then g is computable by an* oaBP *but is is not computable by a read-once DeMorgan formula.*

[8] Notice that for functions sensitive to all variables, read-once DeMorgan formulas are equivalent to general DeMorgan circuits with $n-1$ inner gates. Thus Proposition 6 and Proposition 7 hold for such circuits as well.

Proof. The oaBP for g is a decision tree with the root x, the edge $x = 0$ going to z, the edge $x = 1$ going to y, the edge $y = \alpha$, $z = \alpha$ going to the α-sink for $\alpha \in \{0, 1\}$.

Consider a read-once formula G computing g. Then $G|_{y=0,z=1}$ computes a function $\neg x$, so G must have a leaf $\neg x$. But $G|_{y=1,z=0}$ computes a function x, so G must have a leaf x. That contradicts the read-once property of G. ◀

We note that this example can be extended to a function that is sensitive in an arbitrary number of variables, say for $g(x, y, z) \vee t_1 \cdots \vee t_n$.

2.2 Bipartite Permutation Independent Set Problem

In [9] Ilango showed that Partial MCSP is ETH-hard. The main idea of the proof is to show a reduction from the $(n \times n)$-Bipartite Permutation Independent Set problem that was previously shown to be ETH-hard in [13]. In this section and later we assume that n is even.

In $(n \times n)$-Bipartite Permutation Independent Set problem ($(n \times n)$-BPIS) we are given the adjacency matrix of an undirected graph G over the vertex set $[n] \times [n]$ where every edge is between the sets of vertices $J_1 = [n/2]^2$ and $J_2 = \{n/2 + 1, \ldots, n\}^2$. A graph G is a Yes-instance iff it contains an independent set $S \subseteq J_1 \cup J_2$ of size n such that the coordinates of vertices in S define a permutation of $[n]$, i.e. $\forall i \in [n] \exists j, k \in [n] \colon v = (i, j), w = (k, i), v, w \in S$.

Note that a permutation of $[n]$ corresponding to a Yes-instance of $(n \times n)$-BPIS can be viewed as two permutations on disjoint $n/2$-element subsets. One permutation is defined by $n/2$ vertices from J_1 (corresponding to a permutation of elements $[n/2]$), and another by $n/2$ vertices from J_2 (corresponding to a permutation of elements $\{n/2 + 1, \ldots, n\}$). We use each of these interpretations interchangeably.

▶ **Theorem 8** ([13]). *$(n \times n)$-BPIS cannot be solved in deterministic time $2^{o(n \log n)}$ unless ETH fails.*

Ilango in [9] gave a $2^{O(n)}$-time reduction from $(n \times n)$-BPIS to MCSP*, hence, showing that MCSP* cannot be solved in deterministic time $2^{o(n \log n)}$ unless ETH fails.

3 Main Theorem

In this section we show that the reduction Ilango constructed from ETH-hard problem $(n \times n)$-BPIS to a formula minimization problem holds for the BP minimization problem as well. More precisely we show:

▶ **Theorem 9.** *Suppose that $MBPSP^* \colon \{0, 1, *\}^{2^n} \times [2^n] \to \{0, 1\}$ is in $\mathbf{TIME}[N^{o(\log \log(N))}]$ where $N = 2^n$, then the Exponential Time Hypothesis is false.*

Let us recall the reduction $\mathcal{R} \colon \{0, 1\}^{\binom{n \times n}{2}} \to \{0, 1, *\}^{2^{3n}} \times [2^{3n}]$ from [9][9]. Let $G = ([n] \times [n], E)$ be an instance of $(n \times n)$-BPIS. Then the reduction outputs the pair $(t, 3n - 1)$ where t is the truth table of a partial function $\gamma_G \colon \{0, 1\}^n \times \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1, *\}$ defined as $\gamma_G(\vec{x}, \vec{y}, \vec{z}) =$

[9] We ignore all the edges except for ones between the sets $[n/2] \times [n/2]$ and $\{n/2+1, \ldots n\} \times \{n/2+1, \ldots, n\}$, the second argument is represented in binary form.

$$= \begin{cases} \bigvee_{i\in[n]}(y_i \wedge z_i), & \text{if } \vec{x} = 0^n & (1)\\ \bigvee_{i\in[n]} z_i, & \text{if } \vec{x} = 1^n & (2)\\ \bigvee_{i\in[n]}(x_i \vee y_i), & \text{if } \vec{z} = 1^n & (3)\\ 0, & \text{if } \vec{z} = 0^n & (4)\\ \bigvee_{i=1}^{n/2} x_i & \text{if } \vec{z} = 1^{n/2}0^{n/2} \text{ and } y = 0^n & (5)\\ \bigvee_{i=n/2+1}^{n} x_i & \text{if } \vec{z} = 0^{n/2}1^{n/2} \text{ and } y = 0^n & (6)\\ 1 & \text{if } \exists((j,k),(n/2+j',n/2+k')) \in E\colon (\vec{x},\vec{y},\vec{z}) = (\overline{e_k e_{k'}}, 0^n, e_j e_{j'}) & (7)\\ * & \text{otherwise} & (8) \end{cases}$$

where $\vec{x} = (x_1, \ldots, x_n)$, $\vec{y} = (y_1, \ldots, y_n)$, $\vec{z} = (z_1, \ldots, z_n)$, and $e_i \in \{0,1\}^{n/2}$ is the vector with 1 in the ith entry and zeroes in all the others, and $\overline{e_i}$ is such that $e_i + \overline{e_i} = 1^{n/2}$. In the rest of the proof, we use e_i as a vector with the ith coordinate equal to 1 and all the rest equal to zero with the appropriate dimension.

The intuition behind this reduction is that for every read-once formula, computing this function has to group triples of bits x_i, y_j, and z_k such that $j = k$, and $i \to j$ encodes that an element i maps to an element j in the permutation. The cases (1)-(4) ensure that the read-once formula encodes some permutation on $[n]$. The cases (5) and (6) ensure that the permutation maps the first $n/2$ elements into the first $n/2$ elements and the last $n/2$ elements into the last $n/2$ elements. And, finally, the case (7) enforces that for every edge in the original graph, the permutation defined by the read-once formula contains at most one endpoint of this edge. Note, that every endpoint or vertex chosen in the independent set encodes a mapping of one element of $[n]$ to one element of $[n]$ in the permutation.

As was shown in [9], this reduction maps Yes-instances of BPIS to Yes-instances of $\text{MCSP}^*_{s=m-1}$ where $m = 3n$ is the number of input bits of $\mathcal{R}(G)$:

▶ **Theorem 10** ([9]). *For the reduction $\mathcal{R}$ as above $G \in (n \times n)$-BPIS iff $\mathcal{R}(G) \in MCSP^*_{s=m-1}$.*

To prove Theorem 10, Ilango shows that γ_G extends to a total function computable by a *read-once monotone formula* iff G is a Yes-instance of $(n \times n)$-BPIS. In our work we show that a similar statement holds for *once-appearance branching programs* (oaBP).

▶ **Theorem 11.** *For the reduction $\mathcal{R}$ as above $G \in (n \times n)$-BPIS iff $\mathcal{R}(G)$ can be computed by* oaBP*.*

The plan of the proof is the following:

- In Section 4 we show that the topological order of an oaBP computing γ_G defines a unique permutation $\pi\colon [n] \to [n]$.
- In Section 5 we show that for π as above $\{(i, \pi(i)) \mid i \in [n]\}$ is an independent set in G, thus showing the "only if" direction of the equivalence.
- Since by Proposition 6 that oaBP losslessly simulates read-once formulas, we conclude that Theorem 10 implies the "if" direction of the equivalence.

4 Structure of oaBP Computing γ'

For the first step of our plan it is convenient to simplify the definition of γ_G. Let $\gamma'_n\colon \{0,1\}^{3n} \to \{0,1,*\}$ be equal to γ_G whenever one of (1)-(4) is satisfied, and $*$ otherwise. Here we may choose an arbitrary G as in this domain γ_G does not depend on it.

$$\gamma'_n(\vec{x},\vec{y},\vec{z}) = \begin{cases} \bigvee_{i\in[n]}(y_i \wedge z_i), & \text{if } \vec{x} = 0^n \\ \bigvee_{i\in[n]} z_i, & \text{if } \vec{x} = 1^n \\ \bigvee_{i\in[n]}(x_i \vee y_i), & \text{if } \vec{z} = 1^n \\ 0, & \text{if } \vec{z} = 0^n \\ * & \text{otherwise} \end{cases}$$

We usually shorthand γ'_n as γ'. In Section 4 we identify the structure of an oaBP computing γ'. Let us start by constructing an example of such an oaBP. As we mentioned before, whenever a function is computable by a read-once formula, it is computable by oaBP. Ilango [9] shows that all read-once formulas computing γ' have form

$$\bigvee_{i\in[n]} \left(x_{\pi(i)} \vee (y_i \wedge z_i)\right) \tag{9}$$

for a permutation $\pi \in S_n$. One way to translate this into an oaBP is to chain together n three-node oaBPs each of them computing one disjunct of (9). A topological order of such a oaBP is a sequence of n triplets of $\{x_{\pi(i)}, y_i, z_i\}$ with variables in triplets permuted in some order. We claim that topological order of *every* oaBP computing γ' has this form[10]. We prove this claim in Lemma 12.

We then say that an oaBP B computing γ' *defines* a permutation κ_B mapping $b \mapsto a$ iff $\{x_a, y_b, z_b\} = \{\ell_{3i+1}, \ell_{3i+2}, \ell_{3i+3}\}$ for some $i \in \{0,\dots,n-1\}$. We refer to the triplets of consecutive nodes of form $\{x_a, y_b, z_b\}$ as *xyz-triplets* or *xyz-blocks*.

In this section we prove the following lemma.

▶ **Lemma 12.** *Suppose B is an* oaBP *that computes γ'. Then let $\ell_1, \ell_2, \dots, \ell_{3n}$ be labels of vertices of B in a topological order. Then for every $i \in \{0,..,n-1\}$ we have $\{\ell_{3i+1}, \ell_{3i+2}, \ell_{3i+3}\} = \{x_a, y_b, z_b\}$ for some $a, b \in [n]$.*

The proof of Lemma 12 is surprisingly delicate. The starting point is the characterization of the structure of oaBPs computing the subfunctions of γ' corresponding to the conditions (1)-(4). The main tool in the proofs of these characterizations is the reduction to query complexity, e.g. $\gamma'|_{\vec{x}=0^n}$ has maximum possible query complexity $2n$, which enforces a very tight constraint on the structure of the oaBP computing it. A potential easy proof of Lemma 12 could have followed a logic similar to the proof above if γ' had maximal query complexity $3n$. This, however, is not the case, query complexity[11] of γ' is at most $2n+2$, so such a simple argument is unfortunately ruled out.

4.1 Structure of γ' Conditioned on (1)–(4)

We start by characterizing the structure of oaBP that computes $\gamma'|_{\vec{x}=0^n}$.

[10] Notice that we do not show that the oaBP in question has the exact form as the one constructed according to (9).

[11] Query x_1, x_2, z_1, z_2, if $x_1 \neq x_2$ or $z_1 \neq z_2$, then we may return any value, as none of the conditions (1)-(4) hold. Otherwise, for every possible combination of values x_1, z_1 we have only two possible conditions among (1)-(4) that might hold. If $x_1 = z_1 = \alpha$ we return α right away, if $x_1 = 0$ and $z_1 = 1$, it suffices to query $\vec{y}$ and $\vec{z}$ to get the answer, if $x_1 = 1$ and $z_1 = 0$, it suffices to query $\vec{x}$ and $\vec{z}$ to get the answer.

▶ **Lemma 13.** *The structure of an* oaBP *computing the function* $\bigvee_{i\in[n]}(y_i \wedge z_i)$ *is the following: all nodes except for the* 1*-sink form a single* $2n$*-edge path* $(a_1, b_1, a_2, b_2, \ldots, a_n, b_n, 0\text{-}sink)$ *where* $\{a_i, b_i\} = \{y_{\pi(i)}, z_{\pi(i)}\}$ *for some permutation* $\pi \in S_n$*. For each* $i \in [n]$*, the edge* $a_i = 1$ *goes to* b_i*, the edge* $b_i = 0$ *goes to* a_{i+1}*, the edge* $a_i = 0$ *goes to* a_{i+1} *(for* a_n *it goes to the* 0*-sink), and the edge* $b_i = 1$ *goes to the* 1*-sink.*

Proof. The query complexity of the function $\bigvee_{i\in[n]}(y_i \wedge z_i)$ is $2n$, hence there should exist at least one path of length $2n$ in this branching program. Next, consider the first two variables queried in this path. Assume they do not correspond to y and z with the same label. Without loss of generality assume the first node is labeled by y_1 and the second node is labeled by z_2 (observe that the cases of the first two labels being y_1 and y_2 or z_1 and z_2 are identical, since the variables in the pairs are symmetric). Now, both edges outgoing from y_1 should be parts of the paths that lead to z_2. As for both values of $y_1 = 0$ and $y_1 = 1$ exists a 1-assignment that is zero in all variables other than $z_2 = y_2 = 1$. Hence, if after querying y_1 we do not query z_2, we get a BP that incorrectly computes the function. As we consider the first two nodes of the longest path, we get that both edges from y_1 should directly go to the node labeled by z_2. But it turns the BP to be insensitive to a change in the value of y_1 that contradicts the sensitivity of $\bigvee_{i\in[n]}(y_i \wedge z_i)$ function. Hence the edge $y_1 = 1$ must lead to the node labeled with z_1. Then it remains to observe that after applying the substitutions $y_1 = 0$ or $y_1 = 1, z_1 = 0$ the query complexity of the function decreases by exactly 2. Note that paths corresponding to these two substitutions must lead to the third node in the branching program . This node then must compute $\bigvee_{i\in[n]\setminus\{1\}}(y_i \wedge z_i)$, so the rest of the oaBP has the desired structure by the simple induction on n. ◀

Next, we summarize the structures for the conditions (2)–(4).

▶ **Lemma 14.** *The following structural properties hold.*

(2) *An* oaBP *computing the function* $\gamma'_n|_{\vec{x}=1^n}(\vec{y}, \vec{z}) = \bigvee_{i\in[n]} z_i$ *has the following property. Every node* v *of this* oaBP *reachable from the source computes the function* $\bigvee_{i\in S(v)} z_i$ *where* $S(v) \subseteq [n]$ *is the set of indices of all* z*-variables that do not precede* v *in the topological order.*

(3) *An* oaBP *computing the function* $\gamma'_n|_{\vec{z}=1^n}(\vec{x}, \vec{y}) = \bigvee_{i\in[n]}(x_i \vee y_i)$ *is a Hamiltonian path terminating in the* 0*-sink with edges of the form* $x_i = 0$*,* $y_i = 0$ *going to the next node in the path, and the edges* $x_i = 1$*,* $y_i = 1$ *going to the* 1*-sink.*

(4) *In an* oaBP *computing the function* $\gamma'_n|_{\vec{z}=0^n}(\vec{x}, \vec{y}) \equiv 0$ *the* 1*-sink is not reachable from the source.*

Proof. The structural property for (4) is obvious: if there is a path from the source to the 1-sink, the function is not identically zero.

The structural property for (3) is also easy to see. The query complexity of $\gamma'_n|_{\vec{z}=1^n}$ is $2n$, so the oaBP must have a $2n$-edge path. The edges in these path are the 0-edges and the path terminates in the 0-sink. Suppose an edge of the form $y_i = 1$ or $z_i = 1$ does not go to the 1-sink, say it goes from a node u to a node v, then the path from the source to u according to the Hamiltonian path, then following the edge uv and then continuing to the 0-sink by the Hamiltonian path contradicts the fact that the oaBP computes the disjunction of all variables.

Now let us show the structural property for (2). The source computes the function $\bigvee_{i\in[n]} z_i$ by definition. For any node z_i computing $\bigvee_{j\in S} z_j$ for a set $S \subseteq [n]$ we have that the edge $z_i = 0$ goes to the node computing $\bigvee_{j\in S\setminus\{i\}} z_j$. For any node y_i that is reachable from the source its edges go to the nodes that compute the same function since otherwise

the function computed by the source depends on y_i. Then take any node z_i. Since the function $\gamma'_n|_{\vec{x}=1^n}$ depends on all bits of $\vec{z}$, this node is reachable from the source. Hence, by the observations above, it computes a function of the form $\bigvee_{j\in S} z_j$ with $i \in S$.

Observe that every path from the source to z_i must visit all nodes from $\{z_1, \ldots, z_n\}$ that precede z_i in the topological order. Indeed, if there is a path from the source to z_i that does not visit z_j that precedes z_i in the topological order, then the oaBP returns 0 given the assignment $e_j = 0^{j-1}10^{n-j}$, which is a contradiction. Hence, if the node z_i computes $\bigvee_{j\in S_i} z_j$ and the node z_k computes $\bigvee_{j\in S_k} z_j$, where z_i precedes z_k in the topological order, then $S_i \supseteq S_k$. Moreover $S_i \ni i$ and $S_k \not\ni i$ so $S_i \neq S_k$. Therefore, the sets $S_1, \ldots, S_n$ form a chain by inclusion and are all different, which implies the property given that S_i does not contain indices of nodes that precede z_i in the topological order. ◀

4.2 Proof of Lemma 12

We derive Lemma 12 from the following two lemmas.

▶ **Lemma 15.** *Let B be an* oaBP *computing γ'. Then for every topological ordering of the nodes of B there exists a mapping $\pi\colon [n] \to [n]$ such that $x_{\pi(i)}$ neighbors y_i in the topological order. Additionally, the following property holds*

(†) *if $x_{\pi(i)}$ precedes y_i and z_i in the topological order, then the edge $x_{\pi(i)} = 1$ does not go to 1-sink.*

▶ **Lemma 16.** *Suppose an* oaBP *B computes γ'. Suppose $\ell_1, \ldots, \ell_{3n}$ are the labels of the nodes in some topological ordering of B. Then for every $i \in [3n-4]$ and every $a, b, c \in [n]$ we have $\{\ell_i, \ell_{i+1}, \ldots, \ell_{i+4}\} \neq \{z_a, y_a, x_b, y_c, z_c\}$.*

Proof of Lemma 12. First we show that Lemma 16 guarantees that π given by Lemma 15 is a bijection between $\{y_1, \ldots, y_n\}$ and $\{x_1, \ldots, x_n\}$.

Let $\prec$ denote the topological order of the labels of the nodes in B. Consider a modified version of π that we denote π' such that whenever possible $y_i \prec \pi'(y_i) \prec z_i$ or $z_i \prec \pi'(y_i) \prec y_i$. Observe that this modification never introduces additional violations of injectivity since y_j cannot neighbor a node between y_i and z_i in the topological order for $j \neq i$.

Now suppose $\pi'(y_i) = \pi'(y_j)$ for some $i \neq j$. WLOG assume that $i = 1$ and $j = 2$. Then, by construction of π' and π there exists k such that $\ell_k = y_1, \ell_{k+1} = \pi'(y_1), \ell_{k+2} = y_2$. Notice that $\pi'(y_1)$ must be a single element of $\{x_1, \ldots, x_n\}$ that neighbors y_1 and y_2, thus $\ell_{k-1} = z_1$ and $\ell_{k+3} = z_2$. Then we get a contradiction with the statement of Lemma 16. Hence π' is an injective mapping. As π' is an injective mapping that maps each element of a set of the size n into an element of a set of the size n, we get that π' is a bijection.

By Lemma 13 $\ell_1, \ldots, \ell_{3n}$ has a subsequence of form $a_1, b_1, \ldots, a_n, b_n$ where $\{a_i, b_i\} = \{y_{\tau(i)}, z_{\tau(i)}\}$ for a permutation $\tau\colon [n] \to [n]$. Observe that $\{\ell_1, \ell_2, \ell_3\}$ must contain x_i for $i \in [n]$, otherwise $\{\ell_1, \ell_2, \ell_3\} = \{a_1, b_1, a_2\}$, contradicting Lemma 15 as $y_{\tau(1)}$ does not neighbor $\pi(y_{\tau(1)})$.

If $\{\ell_1, \ell_2, \ell_3\}$ contains more than one element of $\{x_1, \ldots, x_n\}$ then $\{\ell_1, \ell_2, \ell_3\} \ni y_{\tau(1)}$ by surjectivity of π'. Moreover, $\ell_3 = x_b$ and either $\ell_1 = x_a$, $\ell_2 = y_{\tau(1)}$ or $\ell_1 = y_{\tau(1)}$, $\ell_2 = x_a$ for some $a, b \in [n], a \neq b$. Then by Lemma 13 we get $\ell_4 \in \{x_1, \ldots, x_n, z_{\tau(1)}\}$. Thus x_b does not belong to the image of π, which is a contradiction. Therefore, $\{\ell_1, \ell_2, \ell_3\} = \{\pi'(y_{\tau(1)}), y_{\tau(1)}, z_{\tau(1)}\}$. Iteratively applying this procedure to $\{\ell_{3k+1}, \ldots, \ell_{3n}\}$ and π' restricted to $\{y_{\tau(k+1)}, \ldots, y_{\tau(n)}\}$ for $k \in [n]$ concludes the proof of the theorem. ◀

4.3 Proof of Lemma 15

Let B_α for $\alpha \in \{0,1\}$ be the diagram obtained from B by contracting all edges of form $x_i = \alpha$ and removing all edges of form $x_i = 1-\alpha$. Then the diagram B_0 computes $\gamma'|_{\vec{x}=0^n}$. Then from Lemma 13 and (1) we get that B_0 consists of a path of length $2n$ that is split into consequent blocks with two nodes in each, where the labels of the nodes in each block are $\{y_i, z_i\}$ for $i \in [n]$.

Consider an arbitrary block in B_0 and let u and v be the nodes of B corresponding to the nodes in the block, such that u precedes v in the topological order of B.

4.3.1 Case 1: u and v are labeled with z_i and y_i respectively

u must still be present in B_1 and be reachable from the source (say, with the assignment $\vec{x} = \vec{y} = 1^n$; $\vec{z} = 0^n$). Then consider edges in B_0 labeled with $z_i = 1$ and $y_i = 0$: uv and vw, where w is the first node in the subsequent block of B_0, these are uniquely defined since B_0 is an oaBP. We claim that at least one of these edges does not appear in B i.e. it is present in B_0 due to a contraction.

Assume towards a contradiction that both of the edges are present in B. Then let us trace a path from the source of B to the 0-sink according to the assignment $\vec{x} = 1^n$; $\vec{y} = 0^n$; $\vec{z} = e_i$. This path must contain u and w, as z_i-node, corresponding to u should be queried on substitution $x = 0^n$, and then, after assigning $z_i = 1$ and $y_i = 0$ this path leads to the node w. By Lemma 14 for (2) the node w in B_1 and our choice of w, computes $\bigvee_{j \in S} z_j$. Since z_i precedes w in the topological order, $S \not\ni i$. Hence, we get $B(1^n, 0^n, e_i) = 0$ which with value $\gamma'(1^n, 0^n, e_i) = 1$ leads to a contradiction.

Now, since at least one of the edges uv and vw does not appear in B, but appears in B_0, the node v in B is adjacent to a node r labeled with an element of $\{x_1, \ldots, x_n\}$. Since uv and vw belong to the Hamiltonian path in B_0, r must neighbor v in the topological order. Then define $\pi(y_i) = x_j$ where x_j is the label of r. Notice that here the node $x_{\pi(i)}$ never precedes both y_i and z_i, so the property (†) is satisfied.

4.3.2 Case 2: u and v are labeled with y_i and z_i respectively

First, suppose y_i and z_i are the first two nodes in B_0. By Lemma 13 the edge $y_i = 0$ in B_0 goes to the first node of the next yz-pair, let us denote it by w. The edge $y_i = 1$ goes to the z_i-node in B_0. Now, assume, towards a contradiction, that the $y_i = 0$-edge in B goes to w as well. Then, a path in B corresponding to an assignment $s = (1^n, 0^n, e_i)$ skips the node z_i, as y_i is the source of this oaBP, and w is going after z_i in the topological order. Similarly to the proof of Case 1, we get a contradiction with the fact that the subdiagram of B_1 rooted in w computes the function $\bigvee_{j \in S} z_j$ with $S \not\ni i$, which is 0 on s, hence $B(s) = B_1(0^n, e_i) = 0$, whereas $\gamma'(s) = 1$.

Now, assume there is at least one other yz-pair queried before y_i and z_i. Let y_j and z_j be the pair that precedes y_i and z_i in the topological order of B. Note, that in B_0 both edges $y_j = 0$ and $z_j = 0$ go to y_i. And $y_i = 0$ edge in B_0 goes to a node w (skipping z_i), which is the first node in the next yz-block.

▷ Claim 17. One of the edges among $z_j = 0$ and $y_i = 0$ differ in B_0 and B (that is, the edge with this label connects different pairs of nodes).

Proof. The proof of this claim repeats a similar argument in the Case 1. Assume these two edges in B are the same as in B_0. Consider a substitution $s = (1^n, 0^n, e_i)$. As B_1 computes the disjunction of z, a path p corresponding to s in B should query all bits of $\vec{z}$ preceding z_i

in the topological order. Hence, p goes through z_j, and $z_j = 0$ to y_i. Now, as $y_i = 0$ skips z_i and goes to w, which by Lemma 14 computes $\bigvee_{t \in S} z_t$ in B_1. Analogously to the Case 1, $S \not\ni i$, so we get that $B(s) = 0$, which contradicts $\gamma'(s) = 1$. Thus, one of the edges $z_j = 0$ or $y_i = 0$ in B leads to a different node compared to B_0. ◁

4.3.3 Case 2a: the edge $z_j = 0$ goes to y_i

By Claim 17 $y_i = 0$ does not go to w. But in this case it goes to a node r labeled by an x-variable in B. Then as oaBP B queries all bits in $\vec{z}$ on the assignment $(1^n, 0^n, 0^n)$, we get that r goes before z_i in the topological order. So we get that y_i neighbors an x-node in the topological order.

4.3.4 Case 2b: the edge $z_j = 0$ does not go to y_i

Let x_ℓ be the endpoint of $z_j = 0$ in B (notice that $z_j = 0$ can not go to y_j since then it would go there in B_0 as well). Consider the path from x_ℓ to y_i in B according to the assignment $\vec{x} = 0^n$, by the construction of B_0 all nodes on this path are labeled with x-variables. Let x_k be the last node on this path before y_i. We define $\pi(i) = k$. Now it remains to show the following:

1. x_k is the immediate predecessor of y_i in the topological order.
2. The property (†): $x_k = 1$ in B does not go to the 1-sink.

Let us first prove Item 1. Consider the program $B|_{\vec{z}=1^n}$. By Lemma 14 for (3), $B|_{\vec{z}=1^n}$ consists of a single Hamiltonian path of 0-edges. Then, consider the edge $x_k = 0$ in B. This edge goes to y_i, so $x_k y_i$ is present in $B|_{\vec{z}=1^n}$, so in topological order there are no x-node or y-node between x_k and y_i. Given that z_j precedes x_k in the topological order, x_k is the immediate predecessor of y_i.

Now let us prove Item 2. Consider an assignment $\vec{x} = 0^n$, $\vec{y} = 1^n$ and $\vec{z} = 0^n$. Let us trace the path corresponding to this assignment from the root of B to z_j. Notice that z_j is reachable by this path since it is reachable in B_0 by the path according to $\vec{y} = 1^n$, $\vec{z} = 0^n$. After z_j let us follow the edge $z_j = 0$ to x_ℓ and then to x_k following the 0-edges. Suppose that $x_k = 1$ goes to the 1-sink. Let p be the path from the root to the 1-sink that we have traced. Then consider the program $B|_{\vec{z}=0^n}$. Let $p|_{\vec{z}=0^n}$ be the result of contraction of the edges of form $z_t = 0$ in p. $p|_{\vec{z}=0^n}$ exists in $B_{\vec{z}=0^n}$, hence the 1-sink is reachable in this program. This contradicts the fact that $\gamma'|_{\vec{z}=0^n}$ is identically zero.

4.4 Proof of Lemma 16

We are going to gradually derive the structure of the edges going out of the nodes $\{z_a, y_a, x_b, y_c, z_c\}$.

By (3) we have $\gamma'|_{\vec{z}=1^n} = \bigvee_{i \in [n]} x_i \vee y_i$, so by Lemma 14 the edges of from $x_i = 0$ and $y_i = 0$ must either go to a variable in $\{z_1, \ldots, z_n\}$ or to the next variable in $\{x_1, \ldots, x_n, y_1, \ldots, y_n\}$ in the topological order. Hence $y_a = 0$ leads to x_b, $x_b = 0$ leads to y_c.

By (1) we have $\gamma'|_{\vec{x}=0^n} = \bigvee_{i \in [n]} z_i \wedge y_i$, the diagram $B|_{\vec{x}=0^n}$ has structure defined by Lemma 13. Then $z_a = 1$ leads to y_a and $y_c = 1$ leads to z_c and $z_a = 0$ leads either to x_b or y_c. Now observe that the edge $y_a = 1$ goes to the 1-sink by the structure of the diagram $B|_{\vec{z}=1^n}$. If it does go to x_b the assignment $\vec{x} = 0^n$, $\vec{y} = 1^n$ and $\vec{z} = e_a$ is evaluated to 0 by B, which is a contradiction with (3). See Figure 1 for the edges we identified so far.

Now consider an edge $x_b = 1$. By Lemma 14 for (3) it must go to the 1-sink in the diagram $B|_{\vec{z}=1^n}$, so in B it goes either to $\{z_1, \ldots, z_n\}$ or to the 1-sink. Recall that by (2) we have $B|_{\vec{x}=1^n} = \bigvee_{i \in [n]} z_i$. Suppose $x_b = 1$ goes to z_i for $i \in [n]$. Then consider the assignment

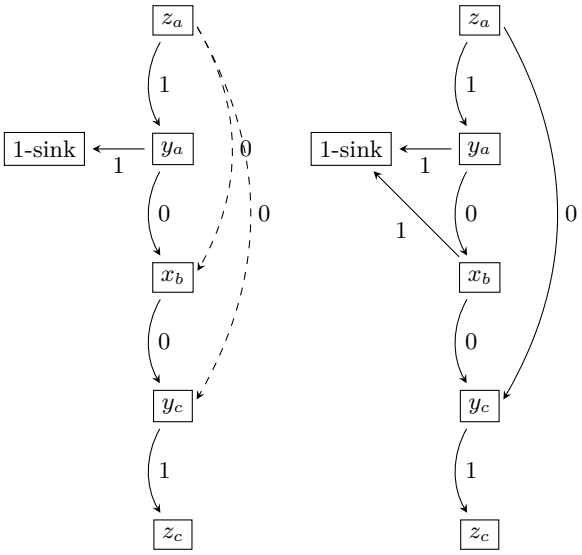


Figure 1 Fixed edges between $\{z_a, y_a, x_b, y_c, z_c\}$.

$\vec{x} = 1^n$, $\vec{z} = e_a$, $\vec{y} = 0^n$. A path p, corresponding to it passes through the node z_a by the structure of $B|_{\vec{x}=1^n}$ given by Lemma 14. Hence, p then leads to x_b and to z_i, which computes $\bigvee_{t\in S} z_t$ for some $S \subseteq [n] \setminus \{a\}$. Thus $B(1^n, 0^n, e_a) = 0$, yet $\gamma'(1^n, 0^n, e_a) = 1$. Hence $x_b = 1$ goes to the 1-sink.

Now we can pinpoint the endpoint of the edge $z_a = 0$. Suppose it goes to x_b. Then we get the contradiction with the assignment $\vec{x} = 1^n$, $\vec{y} = 0^n$ and $\vec{z} = 0^n$. $B(1^n, 0^n, 0^n) = 1$ since we go from z_a to x_b and then to the 1-sink, yet $\gamma(1^n, 0^n, 0^n) = 0$.

Then consider the edge $y_c = 0$, we claim that it goes to z_c. Otherwise consider the assignment $\vec{x} = 1^n$, $\vec{y} = 0^n$ and $\vec{z} = e_c$, $B(1^n, 0^n, e_c) = 0$ since the path of this assignment passes through z_a, then goes to y_c and then to a node strictly below z_c that by Lemma 14 computes $\bigvee_{i\in S} z_i$ for $S \subseteq [n] \setminus \{c\}$. Thus we get a contradiction with $\gamma'(1^n, 0^n, e_c) = 1$.

Then we get the final contradiction as follows: $\gamma'|_{\vec{x}=0^n}$ depends on y_c, but $B|_{\vec{x}=0^n}$ does not, in other words $B(0^n, e_c, e_c) = B(0^n, 0^n, e_c)$, yet $\gamma'(0^n, e_c, e_c) \neq \gamma'(0^n, 0^n, e_c)$.

5 Structure of oaBP Computing γ_G: proof of Theorem 11

The theorem follows from the following lemmas, utilizing the properties (5)-(7) of γ_G.

▶ **Lemma 18.** *For every* oaBP B *computing* γ_G*, the permutation* κ_B *defined by* B *maps elements of* $[n/2]$ *into* $[n/2]$*, and elements of* $\{n/2+1, \ldots, n\}$ *into elements of* $\{n/2+1, \ldots, n\}$.

▶ **Lemma 19.** *For every* oaBP B *computing* γ_G*, the permutation* κ_B *defined by* B *corresponds to an independent set in* G*, that is* $\{(i, \kappa_B(i)) \mid i \in [n]\}$ *is independent in* G.

We start by deriving the theorem from the lemmas.

Proof of Theorem 11. From Lemma 18 and Lemma 19 we get that an oaBP computing γ_G defines a permutation on $[n]$ which maps elements of the first/second half of $[n]$ to the elements of the first/second half of $[n]$ respectively, and for every edge it contains at most one of the vertices of a BPIS instance. Hence, the permutation defined by any oaBP computing γ_G is a correct certificate for a $(n \times n)$-BPIS instance. Thus an oaBP computing γ_G exists iff there is a correct permutation defining independent set in an input $(n \times n)$-BPIS instance. ◀

5.1 Proofs of Lemma 18 and Lemma 19

Both of the lemmas follow from a helper lemma about the structure of oaBP for γ':

▶ **Lemma 20.** *Let B be an* oaBP *computing γ'. Then, if a path p in B goes to the* 1*-sink from one of the nodes in a triplet x_i, z_j, y_j, then p sets at least two variables among x_i, z_j, y_j to* 1*.*

We defer the proof of Lemma 20 to Section 5.2. Now we are ready to prove the lemmas.

Proof of Lemma 18. First, as γ_G is a restricted version of γ', we get that κ_B is well-defined by Lemma 12. Moreover, we know that the topological order of B has the form $a_1, b_1, c_1, a_2, b_2, c_2, \ldots, a_n, b_n, c_n$, where for every $h \in [n]$ there exist $k, \ell \in [n]$, such that $\{a_h, b_h, c_h\} = \{x_k, z_\ell, y_\ell\}$, where $\kappa_B(\ell) = k$.

Let x_i, z_j, y_j form a consequent xyz-triplet in the oaBP, with $i = \kappa_B(j)$. Assume towards a contradiction that i and j are from different halves of the set $[n]$. WLOG let $i \in [n/2]$ and $j \in \{n/2+1, \ldots, n\}$.

Consider a substitution s, in which $z = 1^{n/2}0^{n/2}, y = 0^n, x_i = e_i$. Consider a path p consistent with s. By property (7) it terminates in the 1-sink. Now, consider the last xyz-triplet, that p goes through before getting to the 1-sink. This could either be x_i, z_j, y_j-triplet, or x_k, z_ℓ, y_ℓ, such that $i \neq k; j \neq \ell$. In both cases, at most one of the variables in such xyz-triplets is set to 1. Hence, we get a contradiction with Lemma 20. The proof for the xyz-triplets in the last $n/2$-elements of $[n]$ goes similarly. ◀

Proof of Lemma 19. Consider an edge connecting (j, k) with (j', k') in G. Assume towards a contradiction that κ_B maps an element j to k, and j' to k'. Then, there are two triplets x_j, y_k, z_k, and $x_{j'}, y_{k'}, z_{k'}$ such that within each xyz-triplet the elements are layered consequently in B. Consider a substitution s, which sets $x_j = x_{j'} = 0$, all other bits of $\vec{x}$ to 1, $z_k = z_{k'} = 1$, all other bits of $\vec{z}$ to 0, and all bits of $\vec{y}$ to 0. $\gamma|_s = 1$, hence there is a path p in B consistent with s which goes from the source to 1-sink. Consider a triplet x_a, y_b, z_b, which is queried the last on the path p. Similarly to the previous case we consider two cases, if this triplet is x_j, y_k, z_k (or $x_{j'}, y_{k'}, z_{k'}$, proof for which is analogous) or not. In the latter case, by our assumptions on the structure of B, all $a \neq j$, $a \neq j'$, $b \neq k$, and $b \neq k'$.

If $a = j$ and $b = k$, then, by our choice of s and this triplet, we get that p goes to 1-sink after substituting $x_j = 0, z_k = 1, y_k = 0$. Which is a contradiction with Lemma 20.

Finally, assume either $a \neq j$ and $b \neq k$. Then there is a path in B from the nodes x_a, y_b, z_b directly to 1-sink which is consistent with a substitution $x_a = 1, z_b = 0, y_b = 0$. Thus, again, leads to a contradiction with Lemma 20. ◀

5.2 Proof of Lemma 20

Assume the opposite: there exist paths in B, which go to the 1-sink immediately after substituting exactly one 1-value to variables of an xyz-triplet. Consider a path p among these paths, which goes to the 1-sink from the earliest node according to the topological order of B. Let us denote the xyz-triplet, from which p goes to the 1-sink as x_i, z_j, y_j, and let w be the last non-sink node in p.

▷ Claim 21. p assigns 1 to w.

Proof. Assume the opposite: the last edge in p going to the 1-sink is $w = 0$.

Consider a case $w = x_i$. We want to show that $x_i = 0$-edge cannot lead to the 1-sink. To see this observe that oaBP $B|_{\vec{z}=1^n}$ is sensitive in all x-variables. Hence, in the oaBP $B|_{\vec{z}=1^n}$ a path consistent with setting $\vec{x} = \vec{y} = 0^n$ goes through x_i, and goes to the 0-sink. This contradicts our assumption that the edge $x_i = 0$ goes to 1.

Now consider a case when $w \in \{y_j, z_j\}$. Consider a path p' in the oaBP $B|_{\vec{x}=0^n}$ such that for every $k \in [n]$ $p'(z_k) = 1$ and $p'(y_k) = 0$ if z_k precedes y_k in the topological order, and $p'(z_k) = 0$ and $p'(y_k) = 1$ otherwise. By (1) we have $\gamma'|_{\vec{x}=0^n} = \bigvee_{i\in[n]}(y_i \wedge z_i)$, so every such p' ends in the 0-sink. This contradicts the fact that the edge $w = 0$ goes to the 1-sink, since at least one of such paths p' contains this edge. The claim then follows. ◁

In the remainder of the proof, we show that $w = 1$ cannot be the only 1-value assigned by p to the variables in the last triplet x_i, z_j, y_j. Consider the case when $w = y_j$ or $w = z_j$ and the value of w is assigned to 1 by p. Consider an oaBP $B' = B|_{\vec{x}=0^n}$ corresponding to a partial substitution $\vec{x} = 0^n$. We know the structure of B' from Lemma 13. For each zy-block there is a path to the 1-sink iff we set both $z_i = y_i = 1$. But that contradicts the existence of a path of a form $x_i = 0, y_j = 0, z_j = 1$ or $x_i = 0, y_j = 1, z_j = 0$ from the x_i, z_j, y_j triplet to the 1-sink. Hence, w could not be y or z variable.

Consider the only remaining case, when $w = x_i$, and p sets $x_i = 1$. In the following two claims we show that p should ask either y_j or z_j prior to x_i.

▷ Claim 22. Either y_j or z_j precedes x_i in the topological order.

Proof. Assume the opposite, x_i is the first node in the triplet. By Lemma 15 y_j goes either right before or right after x_i in the topological order. Hence, we get that nodes in this triplet appear in the order x_i, y_j, z_j in the topological order. But then, by the property (†) in Lemma 15, if x_i is the first node of the triplet then the edge $x_i = 1$ does not go to the 1-sink, contradicting the properties of p. ◁

▷ Claim 23. Path p queries either a value of y_j or z_j before querying x_i.

Proof. By Claim 22 we know x_i is not the first node of the triplet x_i, y_j, z_j in the topological order. Now assume that p skips querying yz-nodes preceding x_i in this triplet and goes directly to x_i. Let w' be the last node prior to x_i in p. By our assumption w' belongs to an xyz-triplet $\{x_b, y_a, z_a\}$ different from $\{x_i, y_j, z_j\}$.

First let us rule out the case $w' = z_a$ or $w' = y_a$. If p assigns 0 to w', then the edge $w' = 0$ in $B|_{\vec{x}=0^n}$ goes to the endpoint of the edge $x_i = 0$ in B, which contradicts Lemma 13, since this edge must go to the first variable of the subsequent zy-block, which precedes x_i in the topological order of B. If p assigns 1 to w', then in $B|_{\vec{x}=0^n}$ the edge $w' = 1$ goes to the 1-sink (as it does not go to the node in the same zy-block), hence in B the edge $x_i = 1$ must go to the 1-sink, which implies that the function computed by B does not depend on x_i, which is a contradiction, since γ' does depend on x_i.

Hence $w' = x_b$. Let us now show that p assigns 1 to x_b. Assume the opposite: x_b is set to 0 in B. Let us then trace the path corresponding to the assignment $\vec{x} = 0^n; \vec{z} = 1^n; \vec{y} = 0^n$ in B until we reach the node x_b. This must happen by Lemma 14 for (3). Now by the Lemma 13 and property (1) the edge $x_b = 0$ must go to the first node of the subsequent zy-block, which is a contradiction with the assumption that $x_b = 0$ ends in x_i. Therefore p assigns $x_b = 1$.

Consider an assignment s of $\vec{z} = 1^n, \vec{y} = 0^n$, $\vec{x} = e_b$. By (3) we have $\gamma(s) = 1$. But by our assumptions an edge $x_b = 1$ does not go to the 1-sink immediately, it goes to x_i first. A node x_i should also be reachable by a substitution s' which assigns $\vec{z} = 1^n, \vec{y} = 0^n$, $\vec{x} = 0^n$, as $\gamma'|_{\vec{z}=1^n}$ is sensitive to the value of x_i by (3). But $\gamma(s') = 0$. Hence, we get a contradiction with the fact that paths corresponding to assignments s and s' meet in the node x_i, and follow the same path in B, but one must end in the 0-sink, and another in the 1-sink. Therefore, either z_j or y_j node is not queried by p. ◁

Now, let r be the immediate predecessor of w in p. By Claim 23 we know that either $r \in \{y_j, z_j\}$. First, consider the case of $r = z_j$. Consider a substitution s of $\vec{x} = 1^n$ and $\vec{z} = 0^n$. We know that the oaBP $B|_{\vec{x}=1^n}$ queries all values of z. Hence, $B|_{\vec{x}=1^n}$ contains a path $r = 0, w = 1$, which goes to the 1-sink. But, as the assignment s does not satisfy $B|_{\vec{x}=1^n}$, and is consistent with the substitution $r = 0, w = 1$ we get a contradiction (since the path s passes through r).

Finally, to get a contradiction with the case when $r = y_j$ we show the following two claims.

▷ Claim 24. If $r = y_j$, then z_j goes before y_j and x_i in the topological order.

Proof. Assume the opposite. Let z_j go right after x_i in the topological order. So we get that the order of nodes in this triplet is y_j, x_i, z_j. First, note that $y_j = 0$ edge goes to x_i as $B|_{\vec{z}=1^n}$ queries all ys and xs sequentially, and every edge labeled with 0-edge in $B|_{\vec{z}=1^n}$ goes to either the next node or to the 0-sink. We also know that $x_i = 1$-edge goes to the 1-sink. Now note that that a path consistent with a substitution s defined as $\vec{x} = e_i, \vec{y} = 0^n, \vec{z} = 0^n$ should reach node y_j. But then this path will follow edge $y_j = 0$ to x_i and edge $x_i = 1$ to 1 sink, though $\gamma|_s = 0$. A contradiction. ◁

▷ Claim 25. If z_j is the first node in a xyz-triplet then path p goes through z_j.

Proof. Consider all possible edges which get to this triplet. Let x_a, z_b, y_b be the triplet right before this one. From the structure of $B|_{\vec{x}=1^n}$ we know that all edges from z_b and y_b either go to a node within this triplet, or the 1-sink, or to z_j. We also know that $x_a = 0$ cannot skip z_j, as x_a-node is reachable by a path consistent with a substitution $\vec{x} = 0^n, \vec{y} = 0^n, \vec{z} = 1^n$. Hence if $x_a = 0$ skips z_j then $B|_{\vec{x}=0^n}$ would be insensitive to z_j.

It now remains to show that $x_a = 1$ cannot skip z_j either. In $B|_{\vec{z}=1^n}$ every edge $x_a = 1$ should go to 1-sink. Hence edge $x_a = 1$ either goes to z-node or to 1-sink. This finishes the proof. ◁

From these two claims, we get that p should necessarily go through z_j. Hence, by considering $B|_{\vec{x}=1^n}$ we get that a path consistent $z_j = y_j = 0$ should either go to a node in the next triplet, or go to the 0-sink, in case it is the last triplet of B. Which contradicts the fact that p after substituting $z_j = y_j = 0$ and $x_i = 1$ goes to the 1-sink. This finishes the proof of Lemma 20.

6 Corollaries

Multiple corollaries follow from Theorem 11. For all classes of branching programs that degenerate to an oaBP if constrained on having at most n nodes on an n-bit input, we get that the corresponding minimization problem is ETH-hard. In particular it holds for general branching programs (BP), ordered binary decision diagrams (OBDD) and read-k branching programs for any $k \geq 1$ (k-BP). For definitions of the latter two classes we refer the reader to [27]. Formally we have

▶ **Corollary 26.** *Let $\mathcal{C}$ be a circuit class such that for every $f\colon \{0,1\}^n \to \{0,1\}$ that depends on all its variables there exists $C \in \mathcal{C}$ with $|C| = n$ computing f if and only if there exists* oaBP *B computing f.*

Then, assuming ETH holds we have that $\mathcal{C}$-MCSP$|^*_{s=n}$ requires time $2^{\Omega(n \log n)}$.

It is easy to see that for $\mathcal{C} \in \{\text{OBDD}, \text{BP}\} \cup \{k\text{-BP} \mid k \geq 1\}$ this statement holds, so in particular Theorem 1 is implied.

Proof of Corollary 26. Let $\mathcal{R}$ be the reduction given by Theorem 11. Observe that every partial function in the image of $\mathcal{R}$ can only be extended to a function that depends on all of its variables.

Hence by the assumption on $\mathcal{C}$, $\mathcal{R}(G) \in \mathcal{C}\text{-MCSP}|^*_{s=n}$ iff $\mathcal{R}(G)$ is computable by an oaBP. Therefore, Theorem 11 and Theorem 8 imply the statement. ◀

Following on the results from [6], in which the authors show that $(n \times n)$-BPIS is unconditionally hard for 1-NBPs by showing that reduction of Ilango is computable by a 1-NBPs, we get that similar 1-NBP lower bound holds for all total minimization problems for branching programs.

▶ **Corollary 27.** *For $\mathcal{C}$ as in Corollary 26, the size of the smallest* 1-NBP *that computes $\mathcal{C}$-MCSP is $N^{\Omega(\log\log(N))}$, where N is the input size of $\mathcal{C}$-MCSP.*

Proof. By [6, Theorem 6] we have that the sizes of 1-NBPs computing $\mathcal{C}$-MCSP* and $\mathcal{C}$-MCSP are linearly related (the theorem is only stated for vanilla MCSP, but the proof does not use this). By [6, Threorem 4], the reduction $\mathcal{R}$ can be implemented in 1-NBP, so by Theorem 11 1-NBP size of $\mathcal{C}$-MCSP* with input size $N = 2^{3n}$ is lower bounded by the 1-NBP size of $(n \times n)$-BPIS which by [6, Theorem 3] is $2^{\Omega(n \log n)} = N^{\Omega(\log\log N)}$. ◀

Another corollary follows from the fact that $\mathcal{R}$ always outputs a partial functions that can be encoded as a 2-BP.

▶ **Corollary 28.** *It is* **NP***-hard to check, whether any extension of a partial function expressed by a* 2-BP *over the alphabet $\{0, 1, *\}$ can be computed with an* oaBP*.*

Proof. Since $(n \times n)$-BPIS is an **NP**-complete problem, it suffices to check that for every G the function γ_G can be represented as a 2-BP of polynomial size. First, notice that a 1-BP can distinguish between the cases (1)-(6) and the case that either (7) is true or the function equals to $*$: we read variables in the order $x_1, y_1, z_1, \ldots, x_n, y_n, z_n$ and for each of the patterns among (1)-(6) we maintain if the current prefixes of $\vec{x}$, $\vec{y}$, and $\vec{z}$ coincide with the prefix of the pattern. Thus, at each level of our 1-BP we have 2^6 states. For each of the states at the last level, we choose one of the patterns which is satisfied arbitrary, so we can merge the states into 7: 6 corresponding to the conditions (1)-(6) and one to the case where none of the conditions hold. For each of the cases (1)-(6), the restricted function can easily be computed by a 1-BP.

Now it remains to distinguish the case (7) from a star with a 1-BP. In order to do this we split the input into 6 chunks of length $n/2$ and read then sequentially. For each chunk we maintain the set of possible values among $\{0^{n/2}, 1^{n/2}\} \cup \{e_i \mid i \in [n/2]\} \cup \{\overline{e_i} \mid i \in [n/2]\}$ that the chunk might have given the current prefix. Observe, that the number of potential states is linear. After we read all the chunks we have $O(n^6)$ states at the last level, so we know the only tuple j, k, j', k' that may satisfy the predicate in (7). Since the edges are fixed, we can unite each of the states at the last level with the 1-sink or the $*$-sink. ◀

Finally, to show the **coNP**-hardness of compressing BPs we use the following fact.

▶ **Proposition 29** ([24])**.** *Let $(3, 4)$-SAT be a language of satisfiable* 3-**CNF** *formulas in which every variable occurs in at most 4 clauses. $(3, 4)$-SAT is* **NP***-complete.*

▶ **Corollary 30.** *It is* **coNP***-hard to check whether an input* 4-BP *for a total function can be compressed to a* BP *of a constant size.*

Proof. By Proposition 29 we get that $\overline{(3,4)\text{-SAT}}$ is **coNP**-hard. Now, we construct a reduction from $\overline{(3,4)\text{-SAT}}$ to the partial 4-BP minimization as follows. For a 4-**CNF** formula φ in which every variable appears at most 4 times we construct a branching program B by consequently adding nodes of each clause. Denote the first clause of φ as C_1, and let x_i, x_j, x_k be the variables in it. We add nodes labeled by x_i, x_j, and x_k to B. We set the node labeled by x_i as a source of B. Then, if x_i was negated in C_1 we direct a 1-edge from the x_i-node to the node labeled by x_j, otherwise we direct a 0-edge from x_i to the x_j-node. Then we similarly define one of the edges of x_j to x_k. Finally, if x_k was negated in C_1 we direct its 1-edge to the 1-sink. Then we add nodes from the next clause of φ, and direct all missing edges for nodes x_i, x_j, and x_k to the first node corresponding to the next node. We repeat this procedure for all clauses, redirecting all unassigned edges for nodes corresponding to variables in the last clause of φ to the 0-sink.

Now, for any substitution s of values of the variables B outputs 1 if φ was not satisfied by s, and 0 otherwise. But note, that if φ was an unsatisfiable formula than B would had to output 1 on all inputs. Hence, it could be compressed to a branching program of size 0 (as we count the number of all nodes other than sinks). On the other hand, if φ is satisfiable and has at least one unsatisfying assignment B has to contain at least one node other than a sink. To account for the fact that the size of B corresponding to φ that is satisfiable by any assignment is also 0 (it is just a 0-sink), we change our reduction to first substitute all 0 assignment to φ. If $\varphi(\overline{0}) = 1$ we set B to be a BP for an identity function on one bit which has the size 1. Otherwise, construct B as described before. ◀

Additionally, as for any 3-SAT instance in which every variable appears in at most 4 clauses we can easily construct an equivalent 3-BP of polynomial size. we get that checking whether this BP can compressed to a BP of size 1 allows us to check whether the formula is unsatisfiable.

7 Future Work

The main future direction is to strengthen our results to the total version of MBPSP, and to show **NP**-hardness of MBPSP (or at least MBPSP*). We discuss a couple of barriers for improving results in both of these directions as well as additional open questions motivated by our work.

7.1 Show polynomial-time oracle reduction from MBPSP* to MBPSP

As noted in [10] such reductions are very ad hoc for each specific model. Indeed, the reduction that Ilango showed for the Minimum DeMorgan Formula Size Problem in [10] does not work for branching programs as it relies on the tree-like structure of formulas. The ability of branching programs to reuse the same computation in different branches makes it harder to come up with a suitable reduction. For now we do not even know such a reduction for 1-BPs but the natural next step would be to reduce (1-BP)-MCSP* to (1-BP)-MCSP.

7.2 Prove a stronger conditional lower bound

It would be interesting to find a lower bound stronger than $N^{\Omega(\log\log(N))}$ for the time complexity of MBPSP* (or MFSP*). The obstacle here is that we actually show the ETH-hardness of MBPSP$^*|_n$, for which this time complexity is actually tight, since there are $n^{O(n)} = N^{O(\log\log N)}$ potential branching programs that can compute a function. Hence, in order to have stronger lower bounds we need to consider larger size parameters. The current framework requires us to find explicit functions that require BP-size larger than the

given size parameter. Hence, the best we can hope for without improving the state-of-the-art BP lower bounds is to show that $\text{MBPSP}^*|_{n^2}$ is ETH-hard. Here the trivial upper bound is $n^{O(n^2)} = N^{O(\log N \log\log N)}$, which while still quasipolynomial, is significantly stronger.

7.3 Results for nondeterministic BP minimization

Currently our hardness result only holds for the problem of minimizing deterministic branching programs. But we believe that similar ideas could be used to show that the problem of minimizing nondeterministic branching programs for partial functions is also ETH-hard. Nondeterministic branching programs (NBP) differ from deterministic ones by an additional type of nodes, called *nondeterministic* nodes, that do not have labels. Such nodes correspond to nondeterministic guessing of a branching program. Note, that if we consider a complexity measure that counts *all nodes in an* NBP other than sinks, then our main result immediately extends to the ETH-hardness of minimizing NBP and k-NBP for all k.

However, the standard complexity measure for NBPs is the minimal number of nodes that query a variable (so all nodes other than sinks and nondeterministic nodes) in an NBP computing the function. So one of the approaches to showing the ETH-hardness of the Partial Minimum Nondeterministic Branching Program Size Problem could be in showing that existence of the oaBP with nondeterministic nodes for γ_G is equivalent to the existence of the deterministic oaBP for γ_G. Namely by adding the nondeterministic node we do not make it easier for an oaBP to compute γ_G. Could this result be obtained with an additional careful analysis?

7.4 Could we adapt the NP-hardness proof of Hirahara?

Hirahara in [7] showed a randomized reduction from **NP**-hard problem that is a variation of Minimum Monotone Satisfying Assignment called Collective Minimum Weight Monotone Satisfying Assignment (CMMSA) problem to MCSP^*. One of the possible directions of strengthening our result from the ETH-hardness to **NP**-hardness is by adapting this framework for branching programs. Indeed, the reduction from [7] seems to be robust enough to correctly work on all *no*-instances of CMMSA even for branching programs. But the same reduction does not immediately go through for the *yes*-instances. The main obstacle that we see is in showing an analogue of the Uhlig's lemma [25, 26] for branching programs. This lemma states that in order to compute the direct product of the same function (of arbitrary circuit complexity) on subexponentially many inputs, it is sufficient to use one circuit of size $O(2^n/n)$. Such efficient compression is an important step for constructing a small circuit corresponding to the *yes*-instances of CMMSA. It is not known whether an analogue of Uhlig's lemma exists for branching programs. Rao and Sinha [19] showed a variation of the direct-sum theorem for read-once BPs, hence disproving the analogue of Uhlig's lemma in 1-BP setting. We believe that showing an analogue of Uhlig's lemma for general branching programs in the specific setting required by the Hirahara's proof is a promising direction for showing **NP**-hardness of MBPSP^*.

References

1 Beate Bollig. On the minimization of (complete) ordered binary decision diagrams. *Theory of Computing Systems*, 59(3):532–559, 2016. doi:10.1007/s00224-015-9657-x.

2 Beate Bollig and Ingo Wegener. Improving the variable ordering of obdds is np-complete. *IEEE Trans. Computers*, 45(9):993–1002, 1996. doi:10.1109/12.537122.

3 Magnus Gausdal Find, Alexander Golovnev, Edward A. Hirsch, and Alexander S. Kulikov. A better-than-3n lower bound for the circuit complexity of an explicit function. In *FOCS*, pages 89–98. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.19.

4 S.J. Friedman and K.J. Supowit. Finding the optimal variable ordering for binary decision diagrams. *IEEE Transactions on Computers*, 39(5):710–713, 1990. doi:10.1109/12.53586.

5 Oliver Giel. Branching program size is almost linear in formula size. *Journal of Computer and System Sciences*, 63:222–235, 2001. doi:10.1006/JCSS.2001.1762.

6 Ludmila Glinskih and Artur Riazanov. MCSP is hard for read-once nondeterministic branching programs. In Armando Castañeda and Francisco Rodríguez-Henríquez, editors, *LATIN 2022: Theoretical Informatics - 15th Latin American Symposium, Guanajuato, Mexico, November 7-11, 2022, Proceedings*, volume 13568 of *Lecture Notes in Computer Science*, pages 626–640. Springer, 2022. doi:10.1007/978-3-031-20624-5_38.

7 Shuichi Hirahara. Np-hardness of learning programs and partial MCSP. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 968–979. IEEE, 2022. doi:10.1109/FOCS54457.2022.00095.

8 Shuichi Hirahara, Igor C. Oliveira, and Rahul Santhanam. Np-hardness of minimum circuit size problem for OR-AND-MOD circuits. In Rocco A. Servedio, editor, *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, volume 102 of *LIPIcs*, pages 5:1–5:31. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICS.CCC.2018.5.

9 Rahul Ilango. Constant depth formula and partial function versions of MCSP are hard. In *FOCS*, pages 424–433. IEEE, 2020. doi:10.1109/FOCS46700.2020.00047.

10 Rahul Ilango. The minimum formula size problem is (ETH) hard. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 427–432. IEEE, 2021. doi:10.1109/FOCS52979.2021.00050.

11 Valentine Kabanets and Jin-yi Cai. Circuit minimization problem. In *STOC*, pages 73–79. ACM, 2000. doi:10.1145/335305.335314.

12 Jiatu Li and Tianqi Yang. 3.1n - o(n) circuit lower bounds for explicit functions. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, pages 1180–1193, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3519935.3519976.

13 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Slightly superexponential parameterized problems. *SIAM J. Comput.*, 47(3):675–702, 2018. doi:10.1137/16M1104834.

14 William J. Masek. Some np-complete set covering problems. *Unpublished Manuscript.*, 1979.

15 Cody D. Murray and R. Ryan Williams. On the (non) np-hardness of computing circuit complexity. *Theory of Computing*, 13(4):1–22, 2017. doi:10.4086/toc.2017.v013a004.

16 Eduard I. Nechiporuk. On a boolean function. *Dokl. of USSR Academy of Sciences*, 169(4):765–766, 1966. URL: http://mi.mathnet.ru/dan32449.

17 Arlindo L. Oliveira, Luca P. Carloni, Tiziano Villa, and Alberto L. Sangiovanni-Vincentelli. Exact minimization of binary decision diagrams using implicit techniques. *IEEE Trans. Computers*, 47(11):1282–1296, 1998. doi:10.1109/12.736442.

18 Vaughan R. Pratt. The effect of basis on size of boolean expressions. In *FOCS*, pages 119–121. IEEE Computer Society, 1975. doi:10.1109/SFCS.1975.29.

19 Anup Rao and Makrand Sinha. A direct-sum theorem for read-once branching programs. In Klaus Jansen, Claire Mathieu, José D. P. Rolim, and Chris Umans, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, September 7-9, 2016, Paris, France*, volume 60 of *LIPIcs*, pages 44:1–44:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.APPROX-RANDOM.2016.44.

20 Alexander A. Razborov and Steven Rudich. Natural proofs. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997. doi:10.1006/JCSS.1997.1494.

21 Michael E. Saks and Rahul Santhanam. Circuit lower bounds from np-hardness of MCSP under turing reductions. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 26:1–26:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.CCC.2020.26.

22 Detlef Sieling. The complexity of minimizing and learning obdds and fbdds. *Discrete Applied Mathematics*, 122(1):263–282, 2002. doi:10.1016/S0166-218X(01)00324-9.

23 Yasuhiko Takenaga and Shuzo Yajima. Hardness of identifying the minimum ordered binary decision diagram. *Discret. Appl. Math.*, 107(1-3):191–201, 2000. doi:10.1016/S0166-218X(99)00226-7.

24 Craig A. Tovey. A simplified np-complete satisfiability problem. *Discret. Appl. Math.*, 8(1):85–89, 1984. doi:10.1016/0166-218X(84)90081-7.

25 Dietmar Uhlig. On the synthesis of self-correcting schemes from functional elements with a small number of reliable elements. *Mathematical Notes of the Academy of Sciences of the USSR*, 15.6:558–562, 1974.

26 Dietmar Uhlig. Networks computing boolean functions for multiple input values. *Proceedings of the London Mathematical Society Symposium on Boolean Function Complexity*, pages 165–173, 1992.

27 Ingo Wegener. *Branching Programs and Binary Decision Diagrams*. Society for Industrial and Applied Mathematics, 2000. doi:10.1137/1.9780898719789.

Completeness Theorems for k-SUM and Geometric Friends: Deciding Fragments of Linear Integer Arithmetic

Geri Gokaj
Karlsruhe Institute of Technology, Germany

Marvin Künnemann
Karlsruhe Institute of Technology, Germany

Abstract

In the last three decades, the k-SUM hypothesis has emerged as a satisfying explanation of long-standing time barriers for a variety of algorithmic problems. Yet to this day, the literature knows of only few proven consequences of a refutation of this hypothesis. Taking a descriptive complexity viewpoint, we ask: What is the largest logically defined class of problems *captured* by the k-SUM problem?

To this end, we introduce a class $\mathsf{FOP}_{\mathbb{Z}}$ of problems corresponding to deciding sentences in Presburger arithmetic/linear integer arithmetic over finite subsets of integers. We establish two large fragments for which the k-SUM problem is complete under fine-grained reductions:

1. The k-SUM problem is complete for deciding the sentences with k existential quantifiers.
2. The 3-SUM problem is complete for all 3-quantifier sentences of $\mathsf{FOP}_{\mathbb{Z}}$ expressible using at most 3 linear inequalities.

Specifically, a faster-than-$n^{\lceil k/2\rceil \pm o(1)}$ algorithm for k-SUM (or faster-than-$n^{2\pm o(1)}$ algorithm for 3-SUM, respectively) directly translate to polynomial speedups of a general algorithm for *all* sentences in the respective fragment.

Observing a barrier for proving completeness of 3-SUM for the entire class $\mathsf{FOP}_{\mathbb{Z}}$, we turn to the question which other – seemingly more general – problems are complete for $\mathsf{FOP}_{\mathbb{Z}}$. In this direction, we establish $\mathsf{FOP}_{\mathbb{Z}}$-completeness of the *problem pair* of Pareto Sum Verification and Hausdorff Distance under n Translations under the L_∞/L_1 norm in $\mathbb{Z}^d$. In particular, our results invite to investigate Pareto Sum Verification as a high-dimensional generalization of 3-SUM.

2012 ACM Subject Classification Theory of computation → Complexity theory and logic

Keywords and phrases fine-grained complexity theory, descriptive complexity, presburger arithmetic, completeness results, k-SUM

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.55

Funding This research is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 462679611.

Acknowledgements The authors like to thank the reviewers for constructive feedback as well as Karl Bringmann and Nick Fischer for helpful discussions.

1 Introduction

Consider a basic question in complexity theory: How can we determine for which problems an essentially quadratic-time algorithm is best possible? If a given problem A admits an algorithm running in $n^{2+o(1)}$ time, and it is known that A cannot be solved in time $O(n^{2-\epsilon})$ for any $\epsilon > 0$, then clearly the $n^{2+o(1)}$ algorithm has *optimal* runtime, up to subpolynomial factors. This question can be asked more generally for any $k \geq 1$ and time $n^{k\pm o(1)}$. To this day, the theoretical computer science community is far from able to resolve this question unconditionally. However, a surge of results over recent years uses conditional lower bounds


© Geri Gokaj and Marvin Künnemann;
licensed under Creative Commons License CC-BY 4.0



based on plausible hardness assumptions to shed some light on why some problems seemingly cannot be solved in time $O(n^{k-\epsilon})$ for any $\epsilon > 0$. Most notably, reductions from k-OV, k-SUM and the weighted k-clique problem have been used to establish $n^{k-o(1)}$-time conditional lower bounds, often matching known algorithms; see [50] for a detailed survey.

In this context, the 3-SUM hypothesis is arguably the first – and particularly central – hardness assumption for conditional lower bounds. Initially introduced to explain various quadratic-time barriers observed in computational geometry [35], it has since been used to show quadratic-time hardness for a wealth of problems from various fields [52, 46, 6, 40, 29, 3, 21]. Its generalization, the k-SUM[1] hypothesis, has led to further conditional lower bounds beyond the quadratic-time regime [31, 4, 1, 2, 41]. For a more comprehensive overview, we refer to [50].

The centrality of the 3-SUM hypothesis for understanding quadratic-time barriers begs an interesting question: Does 3-SUM fully capture quadratic-time solvability, in the sense that it is hard for the entire class $\mathsf{DTIME}(n^2)$? Alas, Bloch, Buss, and Goldsmith [10] give evidence that we are unlikely to prove this: If 3-SUM is hard for $\mathsf{DTIME}(n^2)$ under quasilinear reductions, then $\mathsf{P} \neq \mathsf{NP}$. Thus, to understand precisely the role of 3-SUM to understand quadratic-time computation, the more reasonable question to ask is:

What is the largest class $\mathcal{C}$ of problems such that 3-SUM is $\mathcal{C}$-hard?[2]

Finding a large class $\mathcal{C}$ for which 3-SUM is hard may be seen as giving evidence for the 3-SUM hypothesis. Furthermore, such a result may clarify the true expressive power of the 3-SUM hypothesis, much like the NP-completeness of 3-SAT highlights its central role for polynomial intractability.

1.1 Our approach

We approach our central question from a descriptive complexity perspective. This line of research has been initiated by Gao et al. [36], who establish the sparse OV problem as complete for the class of model checking first-order properties. One can interpret this result as showing that the OV problem expresses relational database queries in the sense that a truly subquadratic algorithm for OV would improve the fine-grained data complexity of such queries (see [36] for details). Related works further delineate the fine-grained hardness of model checking first-order properties and related problem classes [49, 14, 12, 7, 13, 32], see Section 1.3 for more discussion.

Towards continuing the line of research on fine-grained completeness theorems, we introduce a class of problems corresponding to deciding formulas in linear integer arithmetic over finite sets of integers. Specifically, consider the vectors

$$x_1 = (x_1[1], \dots, x_1[d_1]), \dots, x_k = (x_k[1], \dots, x_k[d_k])$$

as quantified variables, and let $t_1, \dots, t_l$ be free variables. Moreover, let

$$X := \{x_1[1], \dots, x_1[d_1], \dots x_k[1], \dots, x_k[d_k], t_1, \dots, t_l\},$$

[1] The k-SUM problem asks, given sets $A_1, \dots, A_k$ of n numbers, whether there exist $a_1 \in A_1, \dots, a_k \in A_k$ such that $\sum_{i=1}^{k} a_i = 0$. The k-SUM hypothesis states that for no $\epsilon > 0$ there exists a $O(n^{\lceil k/2 \rceil - \epsilon})$ time algorithm that solves k-SUM.

[2] Note that there are different reasonable notions of reductions to consider. Rather than the quasilinear reductions used by Bloch et al., we will consider the currently more commonly used notion of fine-grained reductions; see Section 1.2 for details on the notion of completeness that we will use.

and let ψ be a quantifier-free linear arithmetic formula over variables in X. We consider the model-checking problem for formulas ϕ in the prenex normal form

$$\phi := Q_1 x_1 \dots Q_k x_k : \psi,$$

where the quantifiers $Q_1, \dots, Q_k \in \{\exists, \forall\}$ are arbitrary. Formally, for such a ϕ, we define the model checking problem $\mathsf{FOP}_{\mathbb{Z}}(\phi)$ as follows[3]

$$\begin{aligned} &\mathsf{FOP}_{\mathbb{Z}}(\phi): \qquad (1)\\ &\quad\textbf{Input:} \text{ Finite sets } A_1 \subseteq \mathbb{Z}^{d_1}, \dots, A_k \subseteq \mathbb{Z}^{d_k} \text{ and } \hat{t_1}, \dots, \hat{t_l} \in \mathbb{Z}.\\ &\textbf{Problem:} \text{ Does } Q_1 x_1 \in A_1 \dots Q_k x_k \in A_k : \psi[(t_1, \dots, t_l)\backslash(\hat{t_1}, \dots, \hat{t_l})] \text{ hold?} \end{aligned}$$

We let $n := \max_i\{|A_i|\}$ denote the input size and will assume throughout the paper that all input numbers (i.e., the coordinates of the vectors in $A_1, \dots, A_k$ and the values $\hat{t_1}, \dots, \hat{t_l}$) are chosen from a polynomially sized universe, i.e., $\{-U, \dots, U\}$ with $U \leq n^c$ for some c. Let $\mathsf{FOP}_{\mathbb{Z}}$ be the union of all $\mathsf{FOP}_{\mathbb{Z}}(\phi)$ problems, where ϕ has at least 3 quantifiers.[4] Besides 3-SUM, a variety of interesting problems is contained in $\mathsf{FOP}_{\mathbb{Z}}$; we discuss a few notable examples below and in the full version.

Frequently, we will distinguish formulas in $\mathsf{FOP}_{\mathbb{Z}}$ using their quantifier structure; e.g., $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\forall)$ describes the class of model checking problems $\mathsf{FOP}_{\mathbb{Z}}(\phi)$ where in ϕ we have $Q_1 = Q_2 = \exists$ and $Q_3 = \forall$. Furthermore, we let $\mathsf{FOP}^k_{\mathbb{Z}}$ be the union of all $\mathsf{FOP}_{\mathbb{Z}}(\phi)$ problems, where ϕ consists of precisely k quantifiers, regardless of their quantifier structure. For a quantifier $Q \in \{\exists, \forall\}$, we write Q^k for the repetition $\underbrace{Q \dots Q}_{k \text{ times}}$. Finally, we remark that a small subset of $\mathsf{FOP}_{\mathbb{Z}}$ has already been studied by An et al. [7], for a discussion see Section 1.3.

1.2 Our Contributions

We seek to determine completeness results for the class $\mathsf{FOP}_{\mathbb{Z}}$. In particular: What are the largest fragments of this class for which 3-SUM (or more generally, k-SUM) is complete? Is there a problem that is complete for the entire class?

Intuitively, we say that a $T_A(n)$-time solvable problem A is *(fine-grained) complete* for a $T_{\mathcal{C}}(n)$-time solvable class of problems $\mathcal{C}$, if the existence of an $O(T_A(n)^{1-\epsilon})$-time algorithm for A with $\epsilon > 0$ implies that for *all* problems C in $\mathcal{C}$ there exists $\delta > 0$ such that C can be solved in time $O(T_{\mathcal{C}}(n)^{1-\delta})$. We extend this notion to completeness of a *family* of problems, since strictly speaking, any (geometric) problem over $\mathbb{Z}^d$ expressible in linear integer arithmetic corresponds to a family of formulas $\mathsf{FOP}_{\mathbb{Z}}$ (one for each $d \in \mathbb{N}$). Formally, consider a family of problems $\mathcal{P}$ with an associated time bound $T_{\mathcal{P}}(n)$ and a class of problems $\mathcal{C}$ with an associated time bound $T_{\mathcal{C}}(n)$; usually $T_{\mathcal{P}}(n), T_{\mathcal{C}}(n)$ denote the running time of the fastest known algorithm solving all problems in $\mathcal{P}$ or $\mathcal{C}$, respectively (often, we omit these time bounds, as they are clear from context).[5] We say that $\mathcal{P}$ is *(fine-grained) complete* for $\mathcal{C}$, if

1. the family $\mathcal{P}$ is a subset of the class $\mathcal{C}$, and

[3] Below, we use the notation $\psi[(t_1, \dots, t_l)\backslash(\hat{t_1}, \dots, \hat{t_l})]$ to denote the substitution of the variables $t_1, \dots, t_l$ by $\hat{t_1}, \dots, \hat{t_l}$ respectively.

[4] It is not too difficult to see that all formulas with 2 quantifiers can be model-checked in near-linear time; see the full version for details.

[5] Here, we use *family* and *class* as a purely semantic and intuitive distinction: A family consists of a small set of similar problems, and a class consists of a large and diverse variety of problems.

2. if for all problems P in $\mathcal{P}$ there exists $\epsilon > 0$ such that P can be solved in time $O(T_{\mathcal{P}}(n)^{1-\epsilon})$, then for all problems C in $\mathcal{C}$ there exists some $\delta > 0$ such that we can solve C in time $O(T_{\mathcal{C}}(n)^{1-\delta})$.

That is, a polynomial-factor improvement for solving the problems in $\mathcal{P}$ would lead to a polynomial-factor improvement in solving *all* problems in $\mathcal{C}$. If a singleton family $\mathcal{P} = \{P\}$ is fine-grained complete for $\mathcal{C}$, then we also say that P is fine-grained complete for $\mathcal{C}$. We work with standard hypotheses and problems encountered in fine-grained complexity; for detailed definitions of these, we refer to the full version of this article.

1.2.1 k-SUM is complete for the existential fragment of $\mathsf{FOP}_{\mathbb{Z}}$

Consider first the existential fragment of $\mathsf{FOP}_{\mathbb{Z}}$, i.e., formulas exhibiting only existential quantifiers. Any $\mathsf{FOP}_{\mathbb{Z}}$ formula with k existential quantifiers can be decided using a standard meet-in-the-middle approach, augmented by orthogonal range search, in time $\tilde{O}(n^{\lceil k/2 \rceil})$[6], see the full version of the paper for details. Since k-SUM is a member of $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$, this running time is optimal up to subpolynomial factors, assuming the k-SUM Hypothesis. As our first contribution, we provide a converse reduction. Specifically, we show that a polynomially improved k-SUM algorithm would give a polynomially improved algorithm for solving the entire class. In our language, we show that k-SUM is fine-grained complete for formulas of $\mathsf{FOP}_{\mathbb{Z}}$ with k existential quantifiers.

▶ **Theorem 1** (k-SUM is $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$-complete)**.** *Let $k \geq 3$ and assume that k-SUM can be solved in time $T_{k\mathrm{SUM}}(n)$. For any problem P in $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$, there exists some c such that P can be solved in time $O(T_{k\mathrm{SUM}}(n) \log^c n)$.*

Thus, if there are $k \geq 3$ and $\epsilon > 0$ such that we can solve k-SUM in time $O(n^{\lceil k/2 \rceil - \epsilon})$, then we can solve all problems in $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ in time $O(n^{\lceil k/2 \rceil - \epsilon'})$ for any $0 < \epsilon' < \epsilon$. By a simple negation argument, we conclude that k-SUM is also complete for the class of problems $\mathsf{FOP}_{\mathbb{Z}}(\forall^k)$.

The above theorem generalizes and unifies previous reductions from problems expressible as $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ formulas to 3-SUM, using different proof ideas: Jafargholi and Viola [39, Lemma 4] give a simple randomized linear-time reduction from triangle detection in sparse graphs to 3-SUM, and a derandomization via certain combinatorial designs. Dudek, Gawrychowski, and Starikovskaya [29] study the family of 3-linear degeneracy testing (3-LDT), which constitutes a large and interesting subset of $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$: This family includes, for any $\alpha_1, \alpha_2, \alpha_3, t \in \mathbb{Z}$, the *3-partite* formula $\exists a_1 \in A_1 \exists a_2 \in A_2 \exists a_3 \in A_3 : \alpha_1 a_1 + \alpha_2 a_2 + \alpha_3 a_3 = t$ and the *1-partite* formula $\exists \alpha_1, \alpha_2, \alpha_3 \in A : \alpha_1 a_1 + \alpha_2 a_2 + \alpha_3 a_3 = t \wedge a_1 \neq a_2 \wedge a_2 \neq a_3 \wedge a_1 \neq a_3$. The authors show that each such formula is either trivial or subquadratic *equivalent* to 3-SUM. For 3-partite formulas, a reduction to 3-SUM is essentially straightforward. For 1-partite formulas, Dudek et al. [29] use color coding.[7] As further examples for reductions from $\mathsf{FOP}_{\mathbb{Z}}$ problems to k-SUM, we highlight a reduction from Vector k-SUM to k-SUM [5] as well as a reduction from $(\min, +)$-convolution to 3-SUM (see [9, 27]) based on a well-known bit-level trick due to Vassilevska Williams and Williams [52], which allows us to reduce inequalities to equalities.

---

[6] We use the notation $\tilde{O}(T) := T \log^{O(1)}(T)$ to hide polylogarithmic factors.

[7] We remark that the reverse direction, i.e., 3-SUM-hardness of non-trivial formulas, is technically much more involved and can be regarded as the main technical contribution of [29].

Perhaps surprisingly in light of its generality and applicability, Theorem 1 is obtained via a very simple, deterministic reduction that combines the tricks from [5, 52]. This generality comes at the cost of polylogarithmic factors (which we do not optimize), which depend on the number of inequalities occurring in the considered formula; for the details see Section 3 and the full version of the paper.

1.2.2 Completeness for counting witnesses

We provide a certain extension of the above completeness result to the problem class of *counting* witnesses to existential $\mathsf{FOP}_{\mathbb{Z}}$ formulas[8]. Counting witnesses is an important task particularly in database applications (usually referred to as model counting). Furthermore, we will make use of witness counting to *decide* certain quantified formulas in subsequent results detailed below. In Section 4, we will obtain the following result.

▶ **Theorem 2.** *Let $k \geq 3$ be odd. If there is $\epsilon > 0$ such that we can count the number of witnesses for k-SUM in time $O(n^{\lceil k/2 \rceil - \epsilon})$, then for all problem P in $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$, there is some $\epsilon' > 0$ such that we can count the number of witnesses for P in time $O(n^{\lceil k/2 \rceil - \epsilon'})$.*

Leveraging the recent breakthrough by [22] that 3-SUM is subquadratic equivalent to counting witnesses of 3-SUM, we obtain the corollary that *3-SUM is hard even for counting witnesses of* $\mathsf{FOP}_{\mathbb{Z}}(\exists^3)$.

▶ **Corollary 3.** *For all problems P in $\mathsf{FOP}_{\mathbb{Z}}(\exists^3)$, there is some $\epsilon_P > 0$ such that we can count the number of witnesses for P in randomized time $O(n^{2-\epsilon_P})$ if and only if there is some $\epsilon' > 0$ such that 3-SUM can be solved in randomized time $O(n^{2-\epsilon'})$.*

1.2.3 Completeness for general quantifier structures of $\mathsf{FOP}_{\mathbb{Z}}$

In light of our first completeness result, one might wonder whether k-SUM is complete for deciding all k-quantifier formulas in $\mathsf{FOP}_{\mathbb{Z}}$, regardless of the quantifier structure of the formulas. Note that for these general quantifier structures, a baseline algorithm with running time $\tilde{O}(n^{k-1})$ can be achieved with a combination of brute-force and orthogonal range queries; see the full version for details.

However, by [7, Theorem 15] there exists a $\mathsf{FOP}_{\mathbb{Z}}(\exists^{k-1}\forall)$-formula ϕ that cannot be solved in time $O(n^{k-1-\epsilon})$-time unless the 3-uniform hyperclique hypothesis is false (see the discussion in Section 1.3). Thus, proving that 3-SUM is complete for all 3-quantifier formulas would establish that the 3-uniform hyperclique hypothesis implies the 3-SUM hypothesis – this would be a novel tight reduction among important problems/hypotheses in fine-grained complexity theory. For $k \geq 4$, it becomes even more intricate: the conditionally optimal running time of $n^{k-1\pm o(1)}$ for $\mathsf{FOP}_{\mathbb{Z}}(\exists^k\forall)$ formulas exceeds the conditionally optimal running time of $n^{\lceil \frac{k}{2} \rceil \pm o(1)}$ for $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ formulas.

We are nevertheless able to obtain a completeness result for general quantifier structures: Specifically, we show that if two geometric problems over $\mathbb{Z}^d$ can be solved in time $O(n^{2-\epsilon_d})$ where $\epsilon_d > 0$ for all d, then each k-quantifier formula in $\mathsf{FOP}_{\mathbb{Z}}$ can be decided in time $O(n^{k-1-\epsilon})$ for some $\epsilon > 0$. These problems are (1) a variation of the Hausdorff distance that we call *Hausdorff distance under n Translations* and (2) the Pareto Sum problem; the details are covered in Section 5.

[8] A witness for a $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ formula $\exists a_1 \in A_1 \dots \exists a_k \in A_k : \varphi$ with $\hat{t_1}, \dots, \hat{t_l} \in \mathbb{Z}$ is a tuple $(a_1, \dots, a_k) \in A_1 \times \cdots \times A_k$ that satisfies the formula $\varphi[(t_1, \dots, t_l) \backslash (\hat{t_1}, \dots, \hat{t_l})]$.

Hausdorff Distance under n Translations

Among the most common translation-invariant distance measures for given point sets B and C is the Hausdorff Distance under Translation [24, 18, 19, 23, 45, 38]. To define it, we denote the directed Hausdorff distance under the L_∞ metric by $\delta_{\overrightarrow{H}}(B,C) := \max_{b\in B}\min_{c\in C}\|b-c\|_\infty$.[9] The Hausdorff distance under translation $\delta^T_{\overrightarrow{H}}(B,C)$ is defined as the minimum Hausdorff distance of B and an arbitrary translation of C, i.e.,

$$\delta^T_{\overrightarrow{H}}(B,C) \coloneqq \min_{\tau\in\mathbb{R}^d}\delta_{\overrightarrow{H}}(B,C+\{\tau\}) = \min_{\tau\in\mathbb{R}^d}\max_{b\in B}\min_{c\in C}\|b-(c+\tau)\|_\infty.$$

For $d=2$, Bringmann et al. [18] were able to show a $(|B||C|)^{1-o(1)}$ time lower bound based on the orthogonal vector hypothesis, and there exists a matching $\tilde{O}(|B||C|)$ upper bound by Chew et al. [25].

We shall establish that restricting the translation vector to be among a set of m candidate vectors yields a central problem in $\mathsf{FOP}_\mathbb{Z}$. Specifically, we define the Hausdorff distance under Translation in A, denoted as $\delta^{T(A)}_{\overrightarrow{H}}(B,C)$, by

$$\delta^{T(A)}_{\overrightarrow{H}}(B,C) \coloneqq \min_{\tau\in A}\delta_{\overrightarrow{H}}(B,C+\{\tau\}) = \min_{\tau\in A}\max_{b\in B}\min_{c\in C}\|b-(c+\tau)\|_\infty.$$

Correspondingly, we define the problem *Hausdorff distance under m Translations* as: Given $A,B,C\subseteq\mathbb{Z}^d$ with $|A|\le m$, $|B|,|C|\le n$ and a distance value $\gamma\in\mathbb{N}$, determine whether $\delta^{T(A)}_{\overrightarrow{H}}(B,C)\le\gamma$. Note that this can be rewritten as a $\mathsf{FOP}_\mathbb{Z}(\exists\forall\exists)$-formula, see the full version of the paper for details.

The Hausdorff distance under m Translations occurs naturally when approximating the Hausdorff distance under translation: Specifically, common algorithms compute a set A of $|A|=f(\epsilon)$ translations such that $\delta^{T(A)}_{\overrightarrow{H}}(B,C)\le(1+\epsilon)\delta^T_{\overrightarrow{H}}(B,C)$. Generally, this problem is then solved by performing $|A|$ computations of the Hausdorff distance, which yields $\tilde{O}(|A|n)=\tilde{O}(f(\epsilon)n)$-time algorithms [48]. Improving over the $\tilde{O}(mn)$-time baseline for Hausdorff Distance under m Translations would thus lead to immediate improvements for approximating the Hausdorff Distance under Translation. Our results will establish additional consequences of fast algorithms for this problem: an $O(n^{2-\epsilon_d})$-time algorithm with $\epsilon_d>0$ for Hausdorff distance under n Translations would give an algorithmic improvement for the classes of $\mathsf{FOP}_\mathbb{Z}(\exists\forall\exists)$- and $\mathsf{FOP}_\mathbb{Z}(\forall\exists\forall)$-formulas.

Verification of Pareto Sums

Our second geometric problem is a verification version of computing *Pareto sums*: Given point sets $A,B\subseteq\mathbb{Z}^d$, the Pareto sum C of A,B is defined as the Pareto front of their sumset $A+B=\{a+b\mid a\in A,b\in B\}$. Put differently, the Pareto sum of A,B is a set of points C satisfying (1) $C\subseteq A+B$, (2) for every $a\in A$ and $b\in B$, the vector $a+b$ is dominated[10] by some $c\in C$ and (3) there are no distinct $c,c'\in C$ such that c' dominates c. The task of computing Pareto sums appears in various multicriteria optimization settings [8, 47, 30, 44]; fast output-sensitive algorithms (both in theory and in practice) have recently been investigated by Hespe, Sanders, Storandt, and Truschel [37].

[9] Since we will exclusively consider the *directed* Hausdorff distance under Translation, we will drop "directed" throughout the paper.

[10] We consider the usual domination notion: A vector $u\in\mathbb{Z}^d$ is dominated by some vector $v\in\mathbb{Z}^d$ (written $u\le v$) if and only if in all dimensions $i\in[d]$ it holds that $u[i]\le v[i]$.

We consider the following problem as *Pareto Sum Verification*: Given $A, B, C \subseteq \mathbb{Z}^d$, determine whether

$$\forall a \in A \forall b \in B \exists c \in C : a + b \leq c.$$

The complexity of Pareto Sum Verification[11] is tightly connected to output-sensitive algorithms for Pareto Sum. Specifically, solving Pareto Sum Verification reduces to *computing* the Pareto sum C when given inputs A, B of size at most n with the promise that $|C| = \Theta(n)$; see Section 7 for details. The work of Hespe et al. [37] gives a practically fast $O(n^2)$-time algorithm in this case for $d = 2$; note that for $d \geq 3$, we still obtain an $\tilde{O}(n^2)$-time algorithm via our Baseline Algorithm, which is described in the full version of the paper.

A problem pair that is complete for $\mathsf{FOP}_{\mathbb{Z}}$

As a pair, these two geometric problems turn out to be fine-grained complete for the class $\mathsf{FOP}_{\mathbb{Z}}$.

▶ **Theorem 4.** *There is a function $\epsilon(d) > 0$ such that both of the following problems can be solved in time $O(n^{2-\epsilon(d)})$*

- *Pareto Sum Verification,*
- *Hausdorff distance under n Translations,*

if and only if for each problem P in $\mathsf{FOP}^k_{\mathbb{Z}}$ with $k \geq 3$ there exists an $\epsilon_P > 0$ such that P can be solved in time $O(n^{k-1-\epsilon_P})$.

The above theorem shows that a single pair of natural problems captures the fine-grained complexity of the expressive and diverse class $\mathsf{FOP}_{\mathbb{Z}}$. As an illustration just how expressive this class is, we observe the following barriers:[12]

1. If there is some $\epsilon > 0$ such that all problems in $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\forall)$ (or $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$) can be solved in time $O(n^{2-\epsilon})$, then OVH (and thus SETH) is false [7, Theorem 16].
2. If there is some $\epsilon > 0$ such that all problems in $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$ (or $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\forall)$) can be solved in time $O(n^{2-\epsilon})$, then the Hitting Set Hypothesis is false [7, Theorem 12].
3. If for all problems P in $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\forall)$ (or $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$), there exists some $\epsilon > 0$ such that we can solve P in $O(n^{2-\epsilon})$, then the 3-uniform Hyperclique Hypothesis is false [7, Theorem 15].
4. If for all problems P in $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$ ($\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\forall)$, $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$, or $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\forall)$), there exists some $\epsilon > 0$ such that we can solve P in time $O(n^{2-\epsilon})$, then the 3-SUM Hypothesis is false (Theorem 1 with Lemma 11).

Theorem 4 raises the question whether for any constant dimension d, the Hausdorff distance under n Translations admits a subquadratic reduction to Pareto Sum Verification. A positive answer would establish Pareto Sum Verification as complete for the *entire* class $\mathsf{FOP}_{\mathbb{Z}}$. We elaborate on this in Section 8.

[11] We remark that our problem definition only checks a single of the three given conditions, specifically, condition (2). However, in Section 7, we will establish that the verifying *all three* conditions reduces to verifying this single condition. More specifically, for sets A, B, C of size at most n, we obtain that if we can solve Pareto Sum Verification in time $T(n)$, then we can check whether C is the Pareto sum of A, B in time $O(T(n))$.

[12] The first three statements follow from $\mathsf{FOP}_{\mathbb{Z}}$ generalizing the class *PTO* studied in [7], see Section 1.3. The remaining statements rely on the additive structure of $\mathsf{FOP}_{\mathbb{Z}}$.

1.2.4 3-SUM is complete for $\mathsf{FOP}_{\mathbb{Z}}$ formulas of low inequality dimension

Returning to our motivating question, we ask: Since it appears unlikely to prove completeness of 3-SUM for all $\mathsf{FOP}_{\mathbb{Z}}$ formulas (as this requires a tight 3-uniform hyperclique lower bound for 3-SUM), can we at least identify a large fragment of $\mathsf{FOP}_{\mathbb{Z}}$ for which 3-SUM is complete? In particular, can we extend our first result of Theorem 1 from existentially quantified formulas to substantially different problems in $\mathsf{FOP}_{\mathbb{Z}}$, displaying other quantifier structures?

Surprisingly, we are able to show that 3-SUM is complete for *low-dimensional* $\mathsf{FOP}_{\mathbb{Z}}$ formulas, *independent of their quantifier structure*. To formalize this, we introduce the *inequality dimension* of a $\mathsf{FOP}_{\mathbb{Z}}$ formula as the smallest number of linear inequalities required to model it. More formally, consider a $\mathsf{FOP}_{\mathbb{Z}}$ formula $\phi = Q_1 x_1 \in A_1, \ldots, Q_k x_k \in A_k : \psi$ with $A_i \subseteq \mathbb{Z}^{d_i}$. The *inequality dimension* of ϕ is the smallest number s such that there exists a Boolean function $\psi' : \{0,1\}^s \to \{0,1\}$ and (strict or non-strict) linear inequalities $L_1, \ldots, L_s$ in the variables $\{x_i[j] : i \in \{1,\ldots,k\}, j \in \{1,\ldots,d_i\}\}$ and the free variables such that $\psi(x_1,\ldots,x_k)$ is equivalent to $\psi'(L_1,\ldots,L_s)$. As an example, the 3-SUM formula $\exists a \in A \exists b \in B \exists c \in C : a + b = c$ has inequality dimension 2, as $a + b = c$ can be modelled as conjunction of the two linear inequalities $a + b \leq c$ and $a + b \geq c$, whereas no single linear inequality can model $a + b = c$.

We show that 3-SUM is fine-grained complete for model-checking $\mathsf{FOP}^3_{\mathbb{Z}}$ formulas with inequality dimension at most 3. This result is our perhaps most interesting technical contribution and intuitively combines our result that 3-SUM is hard for counting $\mathsf{FOP}_{\mathbb{Z}}$ witnesses (Corollary 3) with a geometric argument, specifically, that the union of n unit cubes in $\mathbb{R}^3$ can be decomposed into the union of $O(n)$ pairwise interior- and exterior-disjoint axis-parallel boxes. To this end, we extend a result from [23], which constructs pairwise interior-disjoint axis-parallel boxes, to also achieve exterior-disjointness. For more details, see the Technical Overview below and Section 6.

▶ **Theorem 5.** *There is an algorithm deciding* 3*-SUM in randomized time* $O(n^{2-\epsilon})$ *for an* $\epsilon > 0$*, if and only if for each problem* P *in* $\mathsf{FOP}^k_{\mathbb{Z}}$ *with* $k \geq 3$ *and inequality dimension at most* 3*, there exists some* $\epsilon > 0$ *such that we can solve* P *in randomized time* $O(n^{k-1-\epsilon})$*.*

Note that this fragment of $\mathsf{FOP}_{\mathbb{Z}}$ contains a variety of interesting problems. A general example is given by comparisons of sets defined using the sumset arithmetic[13], which correspond to formulas of inequality dimension at most 2: E.g., checking, given sets $A, B, C \subseteq \mathbb{Z}$ and $t \in \mathbb{Z}$, whether C is an additive t-approximation of the sumset $A + B$ is equivalent to verifying the conjunction of the $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$ problem[14] $A + B \subseteq C + \{0,\ldots,t\}$ and (2) the $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$ problem[15] $C \subseteq A + B$. Likewise, this extends to λ-multiplicative approximations of sumsets. Furthermore, the problems corresponding to general sumset comparisons like $\alpha_1 A_1 + \cdots + \alpha_i A_i \subseteq \alpha_{i+1} A_{i+1} + \cdots + \alpha_k A_k + \{-\ell,\ldots,u\}$ have inequality dimension at most 2 as well.

Our results of Theorems 4 and 5 suggests to view Pareto Sum Verification as a geometric, high-dimensional generalization of 3-SUM. Furthermore, it remains an interesting problem to establish the highest d such that 3-SUM is complete for $\mathsf{FOP}_{\mathbb{Z}}$ formulas of inequality dimension at most d; for a discussion see Section 8.

---

[13] The sumset arithmetic uses the sumset operator $X + Y$ to denote the sumset $\{x + y \mid x \in X, y \in Y\}$ and λX to denote $\{\lambda x \mid x \in X\}$.

[14] Note that the corresponding formula is $\forall a \in A \forall b \in B \exists c \in C : (c \leq a + b) \wedge (a + b \leq c + t)$, which clearly has inequality dimension at most 2.

[15] Note that the corresponding formula is $\forall c \in C \exists a \in A \exists b \in B : a + b = c$, which clearly has inequality dimension at most 2.

Further Applications

As an immediate application of our first completeness theorem, we obtain a simple proof of a $n^{4/3-o(1)}$ lower bound for the 4-SUM problem based on the the 3-uniform hyperclique hypothesis; see the full version of the paper for details. Specifically, by Theorem 1, it suffices to model the 3-uniform 4-hyperclique problem as a problem in $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists\exists)$. The resulting conditional lower bound is implicitly known in the literature, as it can alternatively be obtained by combining a 3-uniform hyperclique lower bound for 4-cycle given in [43] with a folklore reduction from 4-cycle to 4-SUM (see [39] for a deterministic reduction from 3-cycle to 3-SUM).

▶ **Theorem 6.** *If there is some $\epsilon > 0$ such that 4-SUM can be solved in time $O(n^{\frac{4}{3}-\epsilon})$, then the 3-uniform hyperclique hypothesis fails.*

Similarly, we can also give a simple proof for a known lower bound for 3-SUM.

Another application of our results is to establish class-based conditional bounds. As a case in point, consider the problem of computing the Pareto sum of $A, B \subseteq \mathbb{Z}^d$: Clearly, this problem can be solved in time $\tilde{O}(n^2)$ by explicitly computing the sumset $A + B$ and computing the Pareto front using any algorithm running in near-linear time in its input, e.g. [34]. We prove the following conditional optimality results already in the case when the desired output (the Pareto sum of A, B) has size $\Theta(n)$.

▶ **Theorem 7** (Pareto Sum Computation Lower Bound)**.** *The following conditional lower bounds hold for output-sensitive Pareto sum computation:*

1. *If there is $\epsilon > 0$ such that we can compute the Pareto sum C of $A, B \subseteq \mathbb{Z}^2$, whenever C is of size $\Theta(n)$, in time $O(n^{2-\epsilon})$, then the 3-SUM hypothesis fails (thus, for any $\mathsf{FOP}^k_{\mathbb{Z}}$ formula ϕ of inequality dimension at most 3, there is $\epsilon' > 0$ such that ϕ can be decided in time $O(n^{k-1-\epsilon'})$).*
2. *If for all $d \geq 2$, there is $\epsilon > 0$ such that we can compute the Pareto sum C of $A, B \subseteq \mathbb{Z}^d$, whenever C is of size $\Theta(n)$, in time $O(n^{2-\epsilon})$, then there is some $\epsilon' > 0$ such that we can decide all $\mathsf{FOP}_{\mathbb{Z}}$ formulas with k quantifiers not ending in $\exists\forall\exists$ or $\forall\exists\forall$ in time $O(n^{k-1-\epsilon'})$.*

Our lower bound for 2D strengthens a quadratic-time lower bound found by Funke et al. [33] based on the $(\min, +)$-convolution hypothesis to hold already under the weaker (i.e., more believable) 3-SUM hypothesis. For higher dimensions, we furthermore strengthen the conditional lower bound via its connection to $\mathsf{FOP}_{\mathbb{Z}}$.

We conclude with remaining open questions in Section 8.

1.3 Further Related Work

To our knowledge, the first investigation of the connection between classes of model-checking problems and central problems in fine-grained complexity was given by Williams [49], who shows that the k-clique problem is complete for the class of existentially-quantified first order graph properties, among other results. As important follow-up work, Gao et al. [36] establish OV as complete problem for model-checking any first-order property.

Subsequent results include classification results for $\exists^k\forall$-quantified first-order graph properties [14], fine-grained upper and lower bounds for counting witnesses of first-order properties [28], completeness theorems for multidimensional ordering properties [7] (discussed below), completeness and classification results for optimization classes [12, 13] as well as an investigation of sparsity for monochromatic graph properties [32].

We remark that An et al. [7] study completeness results for a strict subset of $\mathsf{FOP}_{\mathbb{Z}}$ formulas: Specifically, they introduce a class $\mathsf{PTO}_{k,d}$ of k-quantifier first-order sentences over inputs $\mathbb{N}^d$ (or, without loss of generality $\{1,\dots,n\}^d$) that may only use *comparisons* of coordinates (and constants). Note that such sentences lack additive structure, and indeed the fine-grained complexity differs decisively from $\mathsf{FOP}_{\mathbb{Z}}$: E.g., for $\mathsf{PTO}(\exists\exists\exists)$ formulas, they establish the sparse triangle detection problem as complete, establishing a conditionally tight running time of $m^{2\omega/(\omega+1)\pm o(1)}$. This is in stark contrast to $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$ formulas, for which we establish 3-SUM as complete problem, yielding a conditionally optimal running time of $n^{2\pm o(1)}$. In particular, for each 3-quantifier structure $Q_1Q_2Q_3$, a $O(n^{2-\epsilon})$-time algorithm for all $\mathsf{FOP}_{\mathbb{Z}}(Q_1Q_2Q_3)$ problems would break a corresponding hardness barrier[16].

Since any $\mathsf{PTO}_{k,d}$ formula is also a $\mathsf{FOP}_{\mathbb{Z}}$ formula with the same quantifier structure, any hardness result in [7] for $\mathsf{PTO}(Q_1,\dots,Q_k)$ carries over to $\mathsf{FOP}_{\mathbb{Z}}(Q_1,\dots,Q_k)$. On the other hand, any of our algorithmic results for $\mathsf{FOP}_{\mathbb{Z}}(Q_1,\dots,Q_k)$ transfers to its subclass $\mathsf{PTO}(Q_1,\dots,Q_k)$.

2 Technical Overview

In this section, we sketch the main ideas behind our proofs.

Completeness of k-SUM for $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$

With the right ingredients, proving that k-SUM is complete for $\mathsf{FOP}_{\mathbb{Z}}$ formulas with k existential quantifiers (Theorem 1) is possible via a simple approach: We observe that any $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ formula ϕ can be rewritten such that we may assume that ϕ is a conjunction of m inequalities. We then use a slight generalization of a bit-level trick of [52] to reduce each inequality to an equality, incurring only $O(\log n)$ overhead per inequality (intuitively, we need to guess the most significant bit position at which the left-hand side and the right-hand side differ). Thus, we obtain $O(\log^m n)$ conjunctions of m equalities; each such conjunction can be regarded as an instance of Vector k-SUM. Using a straightforward approach for reducing Vector k-SUM to k-SUM given in [5], the reduction to k-SUM follows. We give all details in Section 3 and the full version of the paper.

Counting witnesses and handling multisets

While the reduction underlying Theorem 1 preserves the existence of solutions, it fails to preserve the number of solutions. The challenge is that when applying the bit-level trick to reduce inequalities to equalities, we need to make sure that for each witness of a $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ formula ϕ, there is a unique witness in the k-SUM instances produced by the reduction. While it is straightforward to ensure that we do not produce multiple witnesses, the subtle issue arises that distinct witnesses for ϕ may be mapped to the same witness in the k-SUM instances. It turns out that it suffices to solve a *multiset* version of #k-SUM, i.e., to count all witnesses in a k-SUM instance in which each input number may occur multiple times.

[16] Specifically, an $O(n^{2-\epsilon})$ time algorithm for problems in $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists), \mathsf{FOP}_{\mathbb{Z}}(\forall\forall\forall), \mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$, or $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\forall)$ with $\epsilon > 0$ would refute the 3-SUM hypothesis. Furthermore, an $O(n^{2-\epsilon})$ time algorithm for problems in $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$, $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\forall)$, $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$, or $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\forall)$ with $\epsilon > 0$ would immediately yield an improvement for the MaxConv lower bound problem [27]; for details see the full version of the paper.

Thus, to obtain Theorem 2, we show a fine-grained equivalence of Multiset $\#k$-SUM and $\#k$-SUM, for all odd $k \geq 3$. This fine-grained equivalence, which we prove via a heavy-light approach, might be of independent interest.[17] Combining this equivalence with an inclusion-exclusion argument, we may thus lift Theorem 1 to a counting version for all odd $k \geq 3$.

In the reductions below, we will make crucial use of the immediate corollary of Theorem 2 and [22] that for each $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$ formula ϕ, there exists a subquadratic reduction from counting witnesses for ϕ to 3-SUM (Corollary 3).

On general quantifier structures

We perform a systematic study on the different quantifier structures for $k = 3$. Due to simple negation arguments, we only have to perform a systematic study on the classes of problems $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$, $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$, $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$, $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$.

First, we state a simple lemma establishing syntactic complete problems for the classes above.

▶ **Lemma 8** (Syntactic Complete problems (Informal Version)). *Let $Q_1, Q_2 \in \{\exists, \forall\}$. We can reduce every formula of the class $\mathsf{FOP}_{\mathbb{Z}}(Q_1Q_2\exists)$ to the formula*

$$Q_1\tilde{a_1} \in \tilde{A_1}Q_2\tilde{a_2} \in \tilde{A_2}\exists\tilde{a_3} \in \tilde{A_3} : \tilde{a_1} + \tilde{a_2} \leq \tilde{a_3}.$$

On the quantifier change $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists) \to \mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$

We rely on the subquadratic equivalence between 3-SUM and a functional version of 3-SUM called All-ints 3-SUM, which asks to determine for every $a \in A$ whether there is a solution involving a. A randomized subquadratic equivalence was given in [51], which can be turned deterministic [42].

This equivalence allows us to use the bit-level trick to turn inequalities to equalities, despite it seemingly not interacting well with the quantifier structure $\forall\exists\exists$ at first sight. This results in a proof of the following hardness result.

▶ **Lemma 9.** *If 3-SUM can be solved in time $O(n^{2-\epsilon})$ for an $\epsilon > 0$, then all problems P of $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$ can be solved in time $O(n^{2-\epsilon_P})$ for an $\epsilon_P > 0$.*

On the quantifier change $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists) \to \mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$

As a first result for the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$, we are able to show equivalence to 3-SUM for a specific problem in this class, thus introducing a 3-SUM equivalent problem with a different quantifier structure in comparison to 3-SUM. Specifically, we consider the problem of verifying additive t-approximation of sumsets. We are able to precisely characterize the fine-grained complexity depending on t.

Formally, we show the following theorem.

▶ **Theorem 10.** *Consider the Additive Sumset Approximation problem of deciding, given $A, B, C \subseteq \mathbb{Z}, t \in \mathbb{Z}$, whether*

$$A + B \subseteq C + \{0, \ldots, t\}.$$

[17] We remark that it is plausible that the proof of the subquadratic equivalence of 3-SUM and #3-SUM due to Chan et al. [22] could be extended to establish subquadratic equivalence with Multiset #3-SUM as well. Note, however, that a fine-grained equivalence of $\#k$-SUM and k-SUM is not known for any $k \geq 4$.

This problem is

- *solvable in time $O(n^{2-\delta})$ with $\delta > 0$, whenever $t = O(n^{1-\epsilon})$ for any $\epsilon > 0$,*
- *not solvable in time $O(n^{2-\epsilon})$, whenever $t = \Omega(n)$ assuming the Strong 3-SUM hypothesis.*

Furthermore, subquadratic hardness holds under the standard 3-SUM Hypothesis if no restriction on t is made.

The above theorem is essentially enabling a quantifier change transforming the $\exists\exists\exists$ quantifier structure for which 3-SUM is complete into a subquadratic equivalent problem with a quantifier structure $\forall\forall\exists$. Moreover, the 3-SUM hardness is a witness to the hardness of the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$.

Let us remark a few interesting aspects: The algorithmic part follows from sparse convolution techniques going back to Cole and Hariharan [26], see [16] for a recent account and also [20, 17, 15]. Specifically, whenever $t = O(n^{1-\epsilon})$, it holds that $|C + \{0, \dots, t\}| = O(n^{2-\epsilon})$ and intuitively, we can use an output-sensitive convolution algorithm to compute $A + B$ and compare it to $C + \{0, \dots, t\}$.[18] Our result indicates that an explicit construction of $C + \{0, \dots, t\}$ is required, since once it may get as large as $\Omega(n^2)$, we obtain a $n^{2-o(1)}$-time lower bound assuming the Strong 3-SUM Hypothesis.

The lower bound follows from describing the 3-SUM problem alternatively as $(A+B)\cap C \neq \emptyset$, which is equivalent to the negation of $(A + B) \subseteq \bar{C}$, where $\bar{C}$ denotes the complement of C. Thus, we aim to cover the complement of C by intervals of length t. While this appears impossible for 3-SUM, we employ the subquadratic equivalence of 3-SUM and its convolutional version due to Patrascu [46]. This problem will deliver us the necessary structure to represent this complement with the addition of few auxilliary points.

The reverse reduction from Additive Sumset Approximation to 3-SUM follows from Theorem 5 (as Additive Sumset Approximation has inequality dimension 2).

On completeness results for $\mathsf{FOP}^k_{\mathbb{Z}}$

The above ingredients establish our completeness theorems by exhaustive search over remaining quantifiers. Specifically, by a combination of Theorem 10, which shows that Additive Sumset Approximation is 3-SUM hard, and a combination of Lemma 9 and Theorem 1, we get:

▶ **Lemma 11.** *There is a function $\epsilon(d) > 0$ such that the Verification of Pareto Sum problem can be solved in time $O(n^{2-\epsilon(d)})$ if and only if all problems P in the classes*

- $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\exists\exists\exists)$,$\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\forall\forall\forall)$,
- $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\forall\exists\exists)$,$\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\exists\forall\forall)$,
- $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\forall\forall\exists)$,$\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\exists\exists\forall)$,

where $Q_1, \dots Q_{k-3} \in \{\exists, \forall\}$ and $k \geq 3$, can be solved in time $O(n^{k-1-\epsilon_P})$ for an $\epsilon_P > 0$.

Similarly, for quantifier structures ending in $\exists\forall\exists$ and $\forall\exists\forall$, we obtain the following completeness result.

▶ **Lemma 12.** *There is a function $\epsilon(d) > 0$ such that the Hausdorff Distance under n Translations problem can be solved in time $O(n^{2-\epsilon(d)})$ if and only if all problems P in the classes*

- $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\exists\forall\exists)$, $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\forall\exists\forall)$,

where $Q_1, \dots Q_{k-3} \in \{\exists, \forall\}$ and $k \geq 3$, can be solved in time $O(n^{k-1-\epsilon_P})$ for an $\epsilon_P > 0$.

The combination of Lemma 11 and Lemma 12, thus suffice to prove Theorem 4.

[18] The argument is slightly more subtle, since we need to avoid computing $A + B$ if its size exceeds $O(n^{2-\epsilon})$.

The 3-SUM completeness of formulas with inequality dimension at most 3

As a first idea, one could try to solve problems of different quantifier structures by just counting witnesses. Consider in the following the example $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$.

Assume we are promised that the formula $\forall a \in A \forall b \in B \exists c \in C \psi(a,b,c)$ satisfies a kind of *disjointness* property, specifically that for every $(a,b) \in A \times B$ there exists at most one $c \in C$ such that $\psi(a,b,c)$. Then satisfying the formula boils down to checking whether the number of witnesses (a,b,c) satisfiying $\psi(a,b,c)$ equals to $|A| \cdot |B|$.

To create this *disjointness* effect, we use the following geometric approach: We show that one can re-interpret the formula as $\forall a \in A \forall b \in B : a+b \in \bigcup_{c' \in C'} V(c')$, where $A, B, C' \subseteq \mathbb{Z}^3$, C' is a set of size $O(n)$ and $V(c')$ is an orthant associated to c'. Using an adapted variant of [23], we decompose this union of orthants in $\mathbb{R}^3$ (which we may equivalently view as sufficiently large congruent cubes) into a set $\mathcal{R}$ of $O(n)$ *disjoint* boxes. Thus, it remains to notice that the resulting problem – i.e., for all $a \in A, b \in B$ is there a box $R \in \mathcal{R}$ such that $a+b$ is contained in R – is a $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$ formula with the desired disjointness property, which can be handled as argued above.

For the class $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$, we perform a slightly more involved argument. The classes $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$ and $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$ reduce to 3-SUM regardless of the inequality dimension due to Theorem 1 and Lemma 9.

3 k-SUM is complete for existential $\mathsf{FOP}_{\mathbb{Z}}$ formulas

We begin with a simple completeness theorem that k-SUM is complete for the class of problems $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$. Since k-SUM is indeed a $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$-formula, it remains to show a fine-grained reduction from any $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ formula to k-SUM. The proofs in this section are deferred to the full version. As a first step towards this Theorem, we consider how to reduce a conjunction of m linear inequalities to a vector k-SUM instance.

▶ **Lemma 13.** *Consider vectors $a_1 \in \{-U,\dots,U\}^{d_1}, \dots, a_k \in \{-U,\dots,U\}^{d_k}$, integers $S_1,\dots,S_m \in \{-U,\dots,U\}$, for each $i \in \{1,\dots,m\}, j \in \{1,\dots,k\}$, vectors $c_{i,j} \in \mathbb{Z}^{d_j}$, and a formula*

$$\psi := \bigwedge_{i=1}^{m} \left(\sum_{j=1}^{k} c_{i,j}^T a_j \geq S_i \right).$$

There exist $O(1)$ time computable functions $f_1^{\ell,\psi}, \dots, f_k^{\ell,\psi}, g^{\ell,\psi,W}$ such that the following statements are equivalent

1. *The formula $\bigwedge_{i=1}^{m} \left(\sum_{j=1}^{k} c_{i,j}^T a_j \geq S_i \right)$ holds.*
2. *There are $\ell \in \{1,\dots,\lceil \log_2(M) \rceil\}^m, W \in \{1,\dots,k\}^m$ such that $f_1^{\ell,\psi}(a_1) + \cdots + f_k^{\ell,\psi}(a_k) = g^{\ell,\psi,W}(S_1,\dots,S_m)$.*

Moreover, if the second item holds, there is a unique choice of such ℓ and W.

Essentially the above lemma enables a reduction from a conjunction of inequality checks to a conjunction of equality checks. We can now continue with our completeness theorem.

▶ **Theorem 1** (k-SUM is $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$-complete)**.** *Let $k \geq 3$ and assume that k-SUM can be solved in time $T_{k\text{SUM}}(n)$. For any problem P in $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$, there exists some c such that P can be solved in time $O(T_{k\text{SUM}}(n) \log^c n)$.*

Ater applying Lemma 13, it remains to reduce a conjunction of equality checks to k-SUM. To do so, we interpret the conjunction of equalities as a Vector k-SUM problem, which can be reduced to k-SUM in a straightforward way [5].

4 On counting witnesses in $\mathsf{FOP}_{\mathbb{Z}}$

In this section, we show reductions from counting witnesses of $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ formulas to $\#k$-SUM, specifically, we prove Theorem 2. To do so, we adapt the proof of Theorem 1 given in Section 3 to a counting version. As discussed in Section 2, this requires us to work with a multiset version of $\#k$-SUM. Handling multisets is thus the main challenge addressed in this section. Formally, we say that a multiset is a set A together with a function $f : A \to \mathbb{N}$. For $a \in A$, we abbreviate $n_a := f(a)$ as the multiplicity of a. To measure multiset sizes, we still think of each a to have n_a copies in the input, i.e. the size of A is $\sum_{a \in A} n_a$. Almost all proofs in this section are deferred to the full version of the paper.

▶ **Definition 14** ((U,d)-vector Multiset $\#k$-SUM). *Let $X := \{-U, \dots, U\}^d$. Given k multisets $A_1, \dots, A_k \subseteq X$ and $t \in X$, we ask for the total number of k-SUM witnesses, that is*

$$\sum_{\substack{a_1+\dots+a_k=t,\\ a_1 \in A_1, \dots, a_k \in A_k}} \prod_{i=1}^{k} n_{a_i}.$$

Furthermore, define Multiset $\#k$-SUM as $(U,1)$-vector Multiset $\#k$-SUM and M-multiplicity $\#k$-SUM as Multiset $\#k$-SUM with the additional restriction that the multiplicity of each element is limited, that is for all $a \in A_1 \cup \dots \cup A_k : n_a \leq M$ holds. Lastly, $\#k$-SUM is defined as 1-Multiplicity $\#k$-SUM and (U,d)-vector $\#k$-SUM is (U,d)-vector Multiset $\#k$-SUM where for all $a \in A_1 \cup \dots \cup A_k : n_a = 1$ holds.

For the case of $\mathsf{FOP}^3_{\mathbb{Z}}$ we will also introduce the #All-ints version of the above problems, which asks to determine, for each $a_1 \in A_1$, the number of witnesses involving a_1.

The (deferred) proof of the following lemma is analogous to the proof of Abboud et al. [5] to reduce Vector k-SUM to k-SUM.

▶ **Lemma 15** ((U,d)-vector Multiset $\#k$-SUM $\leq_{\lceil k/2 \rceil}$ Multiset $\#k$-SUM). *If Multiset $\#k$-SUM can be solved in time $T(n)$ then (U,d)-vector Multiset $\#k$-SUM can be solved in time $O(nd \log(U) + T(n))$.*

Next, we give a simple approach to solve Multiset $\#k$-SUM when all multiplicities are comparably small.

▶ **Lemma 16** (M-multiplicity $\#k$-SUM$\leq_{\lceil k/2 \rceil}$ $\#k$-SUM). *If $\#k$-SUM can be solved in time $T(n)$, then M-multiplicity $\#k$-SUM can be solved in time $\tilde{O}(T(nM^{k-1}))$.*

For later purposes, we will need the following version of the above lemma.

▶ **Observation 17.** *If #All-ints 3-SUM can be solved in time $T(n)$, then we can solve #All-ints M-multiplicity 3-SUM in time $\tilde{O}(T(nM^2))$.*

We can finally prove the main result of this section.

▶ **Lemma 18.** *For odd $k \geq 3$, if there exists an algorithm for the $\#k$-SUM problem running in time $O(n^{\lceil k/2 \rceil - \epsilon})$ for an $\epsilon > 0$, then there exists an algorithm for the Multiset $\#k$-SUM problem running in time $O(n^{\lceil k/2 \rceil - \epsilon'})$ for an $\epsilon' > 0$.*

Proof. We proceed with a heavy-light approach. Assume there exists an $O(n^{\lceil k/2 \rceil - \epsilon})$ algorithm for the $\#k$-SUM problem. Set $c := (k-1)(\lceil k/2 \rceil)$. Firstly, we count the number of solutions $(a_1, \dots, a_k) \in A_1 \times \dots \times A_k$, where $n_{a_1}, \dots, n_{a_k} \leq n^{\epsilon/c}$ using Lemma 16. This takes time

$$\begin{aligned}\tilde{O}\left((n\cdot(n^{\epsilon/c})^{k-1})^{\lceil k/2\rceil-\epsilon}\right) &= \tilde{O}\left(\left(n^{1+\frac{\epsilon}{\lceil k/2\rceil}}\right)^{\lceil k/2\rceil-\epsilon}\right)\\ &= \tilde{O}\left(n^{\lceil k/2\rceil-\epsilon+\epsilon-\frac{\epsilon^2}{\lceil k/2\rceil}}\right)\\ &= O\left(n^{\lceil k/2\rceil-\epsilon'}\right),\end{aligned}$$

where $\epsilon' > 0$. It remains to calculate the number of witnesses $(a_1, \ldots, a_k)$, where for at least one $i \in \{1, \ldots, k\}$, we have high multiplicity, meaning $n_{a_i} > n^{\epsilon/c}$ holds. Consider the case that $a_1 \in A_1$ is a high-multiplicity number (the case where $a_i \in A_i$ with $i \neq 1$ is a high-multiplicity number is analogous). For each high-multiplicity number a_1 in A_1 we do the following. Solve the $(k-1)$-SUM instance with sets $A_2, \ldots, A_k$ and target $t - a_1$. There are at most $n^{1-(\epsilon/c)}$ many high-multiplicity numbers in A_1, and solving the $(k-1)$-SUM instance takes time $O(n^{(k-1)/2})$, since k is odd. We get a total runtime of

$$\begin{aligned}n^{1-\frac{\epsilon}{c}}\cdot\tilde{O}(n^{(k-1)/2}) &= \tilde{O}(n^{1-(\epsilon/c)+(k-1)/2})\\ &= \tilde{O}(n^{(k+1)/2-(\epsilon/c)})\\ &= O(n^{\lceil k/2\rceil-\epsilon''}),\end{aligned}$$

where $\epsilon'' > 0$, which concludes the proof. ◀

▶ **Theorem 2.** *Let $k \geq 3$ be odd. If there is $\epsilon > 0$ such that we can count the number of witnesses for k-SUM in time $O(n^{\lceil k/2\rceil-\epsilon})$, then for all problem P in $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$, there is some $\epsilon' > 0$ such that we can count the number of witnesses for P in time $O(n^{\lceil k/2\rceil-\epsilon'})$.*

By combining the subquadratic equivalence between 3-SUM and #3-SUM due to Chan et al. [22] and the above theorem, we obtain the following corollary.

▶ **Corollary 3.** *For all problems P in $\mathsf{FOP}_{\mathbb{Z}}(\exists^3)$, there is some $\epsilon_P > 0$ such that we can count the number of witnesses for P in randomized time $O(n^{2-\epsilon_P})$ if and only if there is some $\epsilon' > 0$ such that 3-SUM can be solved in randomized time $O(n^{2-\epsilon'})$.*

The above proof can also be adapted for the special case $k = 3$ to count for each $a_1 \in A_1$ the number of witnesses involving a_1, by plugging in the appropriate All-ints versions; see the full version of the paper for details. Together with the equivalence between #All-ints 3-SUM and 3-SUM of Chan et al. [22], we get

▶ **Corollary 19.** *For all problems P in $\mathsf{FOP}_{\mathbb{Z}}(\exists^3)$, we are able to count for each $a_1 \in A_1$ the number of witnesses involving a_1 in randomized time $O(n^{2-\epsilon})$ for an $\epsilon > 0$, if 3-SUM can be solved in randomized time $O(n^{2-\epsilon'})$ for an $\epsilon' > 0$.*

5 Completeness Theorems for General Quantifier Structures

As Theorem 1 establishes 3-SUM as the complete problem for the class $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$, we would like to similarly explore complete problems for other quantifier structures. All proofs in this section are deferred to the full version. Let us recall our main geometric problems.

▶ **Definition 20** (Verification of d-dimensional Pareto Sum)**.** *Given sets $A, B, C \subseteq \mathbb{Z}^d$. Does the set C dominate $A + B$, that is does for all $a \in A, b \in B$ exist a $c \in C$, with $c \geq a + b$?*

It is easy to see that Verification of d-dimensional Pareto Sum is in $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$.

▶ **Definition 21** (Hausdorff Distance under n Translations). *Given sets $A, B, C \subseteq \mathbb{Z}^d$ with at most n elements and a $\gamma \in \mathbb{N}$, the Hausdorff distance under n Translations problem asks whether the following holds:*

$$\delta_{\overrightarrow{H}}^{T(A)}(B, C) \coloneqq \min_{\tau \in A} \delta_{\overrightarrow{H}}(B, C + \{\tau\}) = \min_{\tau \in A} \max_{b \in B} \min_{c \in C} \|b - (c + \tau)\|_\infty \leq \gamma.$$

We show the following result firstly, which allows us to assume without loss of generality a certain normal form.

▶ **Lemma 22.** *A general $\mathsf{FOP}_{\mathbb{Z}}(Q_1Q_2\exists)$ formula, with input set $A_1 \subseteq \mathbb{Z}^{d_1}, A_2 \subseteq \mathbb{Z}^{d_2}, A_3 \subseteq \mathbb{Z}^{d_3}$, where $|A_1| = |A_2| = |A_3| = n$, can be reduced to the $\mathsf{FOP}_{\mathbb{Z}}(Q_1Q_2\exists)$ formula*

$$Q_1 a_1' \in A_1' Q_2 a_2' \in A_2' \exists a_3' \in A_3' : a_1' + a_2' \leq a_3'$$

in time $O(n)$, where $|A_1'| = |A_2'| = n$ and $|A_3'| = O(n)$.

The above lemma immediately gives us complete syntactic problems for our classes. It remains to establish connections between the different quantifier structure classes, and explore natural variants of the syntactic problems.

The syntactic complete problem for the class $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$ turns out to be equivalent to Hausdorff Distance under n Translations. We obtain:

▶ **Lemma 23** (Hausdorff Distance under n Translations is complete for $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$). *There is a function $\epsilon(d) > 0$ such that Hausdorff Distance under n Translations can be solved in time $O(n^{2-\epsilon(d)})$ if and only if all problems P in $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$ can be solved in time $O(n^{2-\epsilon_P})$ for an $\epsilon_P > 0$.*

Similarly, the Verification of Pareto Sum problem is complete for the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$.

▶ **Lemma 24** (Verification of Pareto Sum is complete for $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$). *There is a function $\epsilon(d) > 0$ such that Verification of Pareto Sum can be solved in time $O(n^{2-\epsilon(d)})$ if and only if all problems P in $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$ can be solved in time $O(n^{2-\epsilon_P})$ for an $\epsilon_P > 0$.*

5.1 $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists) \to \mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$

We continue with handling the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$. By simply making use of Corollary 19, one can easily prove that 3-SUM is hard for the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$. We can also show a deterministic proof, as Corollary 19 makes use of the subquadratic equivalence between 3-SUM and #All-ints 3-SUM, which relies on randomization techniques.

▶ **Lemma 9.** *If 3-SUM can be solved in time $O(n^{2-\epsilon})$ for an $\epsilon > 0$, then all problems P of $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$ can be solved in time $O(n^{2-\epsilon_P})$ for an $\epsilon_P > 0$.*

5.2 $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists) \to \mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$

We explore the connection between the problem Additive Sumset Approximation, which is a member of the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$, and the 3-SUM problem. The following theorem will play a key role to enable the discovery of the relationship between 3-SUM and other quantifier structures.

▶ **Theorem 10.** *Consider the Additive Sumset Approximation problem of deciding, given* $A, B, C \subseteq \mathbb{Z}, t \in \mathbb{Z}$*, whether*

$$A + B \subseteq C + \{0, \dots, t\}.$$

This problem is

- *solvable in time* $O(n^{2-\delta})$ *with* $\delta > 0$*, whenever* $t = O(n^{1-\epsilon})$ *for any* $\epsilon > 0$*,*
- *not solvable in time* $O(n^{2-\epsilon})$*, whenever* $t = \Omega(n)$ *assuming the Strong 3-SUM hypothesis.*

Furthermore, subquadratic hardness holds under the standard 3-SUM Hypothesis if no restriction on t *is made.*

The proof is deferred to the full version of the paper.

5.3 Completeness results for the class $\mathsf{FOP}^k_{\mathbb{Z}}$

We turn to combining the above insights to establish (a pair of) complete problems for the class $\mathsf{FOP}_{\mathbb{Z}}$. The proofs in this section are deferred to the full version of the paper.

▶ **Lemma 11.** *There is a function* $\epsilon(d) > 0$ *such that the Verification of Pareto Sum problem can be solved in time* $O(n^{2-\epsilon(d)})$ *if and only if all problems* P *in the classes*

- $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\exists\exists\exists)$,$\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\forall\forall\forall)$,
- $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\forall\exists\exists)$,$\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\exists\forall\forall)$,
- $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\forall\forall\exists)$,$\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\exists\exists\forall)$,

where $Q_1, \dots Q_{k-3} \in \{\exists, \forall\}$ *and* $k \geq 3$*, can be solved in time* $O(n^{k-1-\epsilon_P})$ *for an* $\epsilon_P > 0$*.*

▶ **Lemma 12.** *There is a function* $\epsilon(d) > 0$ *such that the Hausdorff Distance under* n *Translations problem can be solved in time* $O(n^{2-\epsilon(d)})$ *if and only if all problems* P *in the classes*

- $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\exists\forall\exists), \mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\forall\exists\forall)$,

where $Q_1, \dots Q_{k-3} \in \{\exists, \forall\}$ *and* $k \geq 3$*, can be solved in time* $O(n^{k-1-\epsilon_P})$ *for an* $\epsilon_P > 0$*.*

We finally obtain our completeness theorem for the whole class $\mathsf{FOP}^k_{\mathbb{Z}}$.

▶ **Theorem 4.** *There is a function* $\epsilon(d) > 0$ *such that both of the following problems can be solved in time* $O(n^{2-\epsilon(d)})$

- *Pareto Sum Verification,*
- *Hausdorff distance under* n *Translations,*

if and only if for each problem P *in* $\mathsf{FOP}^k_{\mathbb{Z}}$ *with* $k \geq 3$ *there exists an* $\epsilon_P > 0$ *such that* P *can be solved in time* $O(n^{k-1-\epsilon_P})$*.*

Essentially, these two problems capture the complexity of the class $\mathsf{FOP}^3_{\mathbb{Z}}$ and can be seen as the most important problems in $\mathsf{FOP}^k_{\mathbb{Z}}$.

6 The 3-SUM problem is complete for $\mathsf{FOP}_{\mathbb{Z}}$ formulas with Inequality Dimension at most 3

In this section, we show that 3-SUM problem captures an interesting subclass of $\mathsf{FOP}_{\mathbb{Z}}$ formulas with arbitrary quantifier structure, namely the formulas of sufficiently small *inequality dimension*. Let us recall the notion of inequality dimension.

▶ **Definition 25** (Inequality Dimension of a Formula). *Let $\phi = Q_1 x_1 \in A_1, \ldots, Q_k x_k \in A_k : \psi$ be a $\mathsf{FOP}_{\mathbb{Z}}$ formula with $A_i \subseteq \mathbb{Z}^{d_i}$.*

The inequality dimension *of ϕ is the smallest number s such that there exists a Boolean function $\psi' : \{0,1\}^s \to \{0,1\}$ and (strict or non-strict) linear inequalities $L_1, \ldots, L_s$ in the variables $\{x_i[j] : i \in \{1, \ldots, k\}, j \in \{1, \ldots, d_i\}\}$ and the free variables such that $\psi(x_1, \ldots, x_k)$ is equivalent to $\psi'(L_1, \ldots, L_s)$.*

In the following, we look at the class of problems $\mathsf{FOP}^k_{\mathbb{Z}}$ with the restriction of inequality dimension at most 3. We use the following naming convention for boxes.

▶ **Definition 26.** *A d-box in $\mathbb{R}^d$ is the cartesian product of d proper intervals $s_1 \times \cdots \times s_d$, where s_i is an open, closed or half-open interval. We call a cartesian product of only closed intervals a closed box and a cartesian product of only open intervals an open box.*

Given a set R of n closed boxes (represented as $2d$ integer coordinates), and d-dimensional points $a \in A, b \in B$, we can express in $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$ whether $a + b$ lies in one of the boxes as follows:

$$\exists a \in A \exists b \in B \exists r \in R : \bigwedge_{i=1}^{d} r[i] \leq a[i] + b[i] \wedge a[i] + b[i] \leq r[d+i].$$

In fact, we are not limited to closed boxes, if a box is open or half open in a dimension, one can adjust the inequalities in this dimension appropriately.

In order to prove our main theorem in this section, we need to partition the union of n unit cubes in $\mathbb{R}^3$ into pairwise interior- and exterior-disjoint boxes. While Chew et al. [23] studied such a decomposition of unit cubes with the requirement of only interior-disjoint boxes, we need an extension of their result to guarantee disjoint exteriors.

▶ **Lemma 27** (Disjoint decomposition of the union of cubes in $\mathbb{R}^3$). *Let $\mathcal{C}$ be a set of n axis-aligned congruent cubes in $\mathbb{R}^3$. The union of these cubes, can be decomposed into $O(n)$ boxes whose interiors and exteriors are disjoint in time $O(n \log^2 n)$.*

The proof is deferred to the full version.

▶ **Theorem 28.** *There is an algorithm deciding 3-SUM in randomized time $O(n^{2-\epsilon})$ for an $\epsilon > 0$ if and only if for each problem P in the classes $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$ and $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$ of inequality dimension at most 3 there exists some $\epsilon' > 0$ such that we can solve P in randomized time $O(n^{2-\epsilon'})$.*

Proof. For the first direction due to Theorem 10, we can reduce 3-SUM to an instance of Additive Sumset Approximation,

$$\forall a \in A \forall b \in B \exists c \in C : c \leq a + b \wedge a + b \leq c + t,$$

which has inequality dimension 2. Let us continue with the other direction. Let $\phi := Q_1 a \in A \forall b \in B \exists c \in C : \varphi$, where $Q_1 \in \{\exists, \forall\}$ and φ is a quantifier free linear arithmetic formula with inequality dimension 3. Let $L_1 := \alpha_1^T a + \beta_1^T b \leq \gamma_1^T c + S_1$, $L_2 := \alpha_2^T a + \beta_2^T b \leq \gamma_2^T c + S_2$ and $L_3 := \alpha_3^T a + \beta_3^T b \leq \gamma_3^T c + S_3$ after replacing the free variables. Assume that the formula φ is given in DNF, thus each co-clause has at most 3 atoms, chosen from L_1, L_2, L_3 and their negations. Let

$$A' := \left\{ \begin{pmatrix} \alpha_1^T a \\ \alpha_2^T a \\ \alpha_3^T a \end{pmatrix} : a \in A \right\}, B' := \left\{ \begin{pmatrix} \beta_1^T b \\ \beta_2^T b \\ \beta_3^T b \end{pmatrix} : b \in B \right\}, C' := \left\{ \begin{pmatrix} \gamma_1^T c + S_1 \\ \gamma_2^T c + S_2 \\ \gamma_3^T c + S_3 \end{pmatrix} : c \in C \right\}$$

Thus each co-clause consists of conjunctions of a subset of the following set

$$\{a'[0]+b'[0] \le c'[0], a'[0]+b'[0] \ge c'[0]+1, a'[1]+b'[1] \le c'[1],$$
$$a'[1]+b'[1] \ge c'[1]+1, a'[2]+b'[2] \le c'[2], a'[2]+b'[2] \ge c'[2]+1\}.$$

Let the co-clauses of φ be $V_1, \dots, V_h$. Thus, we aim to decide a formula of the form:

$$Q_1 a' \in A' \forall b' \in B' \exists c' \in C' : \bigvee_{i=1}^{h} V_i \tag{2}$$

For each co-clause V_i, $i \in \{1, \dots, h\}$ it holds that V_i is of the form

$$\bigwedge_{k \in V_i^K} L_k \wedge \bigwedge_{j \in V_i^J} \neg L_j,$$

for some $V_i^J, V_i^K \subseteq \{1,2,3\}$ and $V_i^J \cap V_i^K = \emptyset$.

Let us consider for each fixed $c' \in C'$ the following possibly empty orthant in $\mathbb{R}^3$.

$$\mathcal{S}(V_i, c') := \{x \in \mathbb{R}^3 : \bigwedge_{k \in V_i^K} x[k] \le c'[k] \wedge \bigwedge_{j \in V_i^J} x[j] \ge c'[j]+1\}.$$

By construction, it is immediate that for a fixed c' and $(a', b') \in A' \times B'$ that (a', b', c') fulfill the co-clause V_i if and only if $a'+b' \in \mathcal{S}(V_i, c')$. Thus, equivalently to (2), we ask

$$Q_1 a' \in A' \forall b' \in B' \exists c' \in C' : \bigvee_{i=1}^{h} \left(a'+b' \in S(V_i, c')\right).$$

Having a closer look, $\bigvee_{i=1}^{h} (a'+b' \in S(V_i, c'))$ is true if and only if $a'+b'$ lies in one of the orthants $S(V_i, c')$.

We argue that we may represent the orthant $\mathcal{S}(V_i, c')$ as an appropriately chosen cube in $\mathbb{R}^3$. To this end, let $M := 2 \cdot \max\{\|a\|_1 + \|b\|_1 + \|c\|_1 : a' \in A', b' \in B', c' \in C'\}$ be a sufficiently large number. We can interpret $\mathcal{S}(V_i, c')$ as a cube of the type $\mathcal{C}_{i,c'} = [m_0, m_0'] \times [m_1, m_1'] \times [m_2, m_2']$, where for $u \in \{0,1,2\}$, we define:

$$m_u := \begin{cases} -M & u \notin V_i^K, u \notin V_i^J, \\ -2M + c[u] & u \in V_i^K, \\ c[u]+1 & u \in V_i^J, \end{cases} \qquad m_u' := \begin{cases} M & u \notin V_i^K, u \notin V_i^J, \\ c[u] & u \in V_i^K, \\ 2M + c[u] + 1 & u \in V_i^J. \end{cases}$$

The cubes are axis-aligned and have side length $2M$. Due to the large size of the cube we get for fixed $c' \in C'$ that $a'+b' \in \mathcal{S}(V_i, c')$ if and only if $a'+b'$ lies inside the cube $\mathcal{C}_{i,c'}$.

By Lemma 27, we can decompose the collection of cubes $\mathcal{C}_{i,c'}$ for $i \in \{1, \dots, H\}, c' \in C'$ into $l = O(n)$ disjoint boxes $\mathcal{R} := \{R_1, \dots, R_l\}$ in time $O(n \log^2 n)$. Let us now go through a case distinction based on the first quantifier.

- If $Q_1 = \forall$, equivalent to ϕ we ask

  $$\forall a' \in A' \forall b' \in B' \exists i \in \{1, \dots, l\} : a'+b' \text{ lies in } R_i.$$

  By replacing each $i \in \{1, \dots, l\}$ by a 6-tuple denoting the dimensions of the box R_i, we can reduce counting the number of (a', b', R_i) with $a'+b' \in R_i$ to 3-SUM using Corollary 3. Due to the disjointness of the boxes R_i, we know that no (a', b') can be in different boxes $R_i, R_{i'}$ with $i \neq i'$.

  Thus, we can decide our original question by checking whether the number of such witnesses equals $|A'| \cdot |B'|$, concluding the fine-grained reduction to 3-SUM.

- Assume now that $Q_1 = \exists$. Thus, equivalently to ϕ, we ask.

  $$\exists a' \in A' \forall b' \in B' \exists i \in \{1, \ldots, l\} : a' + b' \text{ lies in } R_i.$$

  We can now make use of Corollary 19. Count for each $a' \in A'$ the number of *witnesses* (a', b', R_i) with $a' + b' \in R'$. We claim that it remains to check whether there is some a' that is involved in $|B'|$ witnesses. To see this, note that due to the disjointness of the R_i's, for any $a' \in A'$ we have that the number of (b', R_i) with $a' + b' \in R_i$ is equal to the number of b' such that there exists R_i with $a' + b' \in R_i$. Again, the desired reduction to 3-SUM follows. ◀

We remark that, by [11], we know that the complexity of the union of orthants in $\mathbb{R}^d$ has worst case complexity $O(n^{\lfloor d/2 \rfloor})$. Thus, the above proof does not seem directly generalizable to inequality dimensions larger than 3. We can extend Theorem 28 to k-quantifiers by the following theorem.

▶ **Theorem 5.** *There is an algorithm deciding 3-SUM in randomized time $O(n^{2-\epsilon})$ for an $\epsilon > 0$, if and only if for each problem P in $\mathsf{FOP}^k_{\mathbb{Z}}$ with $k \geq 3$ and inequality dimension at most 3, there exists some $\epsilon > 0$ such that we can solve P in randomized time $O(n^{k-1-\epsilon})$.*

The above theorem gives us immediate reductions to 3-SUM for many seemingly unrelated problems of different quantifier structures and semantics.

For instance, as a direct application of the above theorem we can conclude the equivalence of the Additive Sumset Approximation problem to 3-SUM, together with Theorem 10.

▶ **Lemma 29** (Additive Sumset Approximation $\leq_2$ 3-SUM)**.** *If the 3-SUM problem can be solved in randomized time $O(n^{2-\epsilon})$ for an $\epsilon > 0$ then Additive Sumset Approximation problem can be solved in randomized time $O(n^{2-\epsilon'})$ for an $\epsilon' > 0$.*

7 Application: A lower bound on the computation of Pareto Sums

In the following, we explore how the 3-SUM hardness of Verification of Pareto Sum translates to a hardness result for the problem of computing Pareto Sums. Let us first justify the naming of the Verification of Pareto Sum problem, by showing it to be subquadratic equivalent to the more natural extended version of Verification of Pareto Sum. Throughout this section, we consider dimensions $d \geq 2$.

▶ **Definition 30** (Verification of Pareto Sum (Extended version))**.** *Given sets $A, B, C \subseteq \mathbb{Z}^d$, do the following properties hold simultaneously:*

- *(Inclusion): $C \subseteq A + B$,*
- *(Dominance): C dominates $A + B$. More formally, for every $a \in A, b \in B$ there exists $c \in C$ with $c \geq a + b$.*
- *(Minimality): There are no $c, c' \in C$ with $c \neq c'$ and $c \leq c'$.*

We make use of the following lemma and its construction for the results in this section.

▶ **Lemma 31.** *Given sets $A, B, C \subseteq \mathbb{Z}^d$ of size at most n, one can construct sets $\tilde{A}, \tilde{B}, \tilde{C} \subseteq \mathbb{Z}^d$ of size $\Theta(n)$ in time $\tilde{O}(n)$ such that (1) $\tilde{A}, \tilde{B}, \tilde{C}$ always satisfy the minimality and inclusion condition and (2) $\tilde{A}, \tilde{B}, \tilde{C}$ fulfill the dominance condition if and only if A, B, C fulfill the dominance condition.*

Due to space constraints, the proof had to be deferred to the full version. Using this construction, it is not difficult to obtain the following equivalence.

▶ **Lemma 32.** *There is an $O(n^{2-\epsilon})$ time algorithm for an $\epsilon > 0$ for Verification of Pareto Sum (Extended Version) if and only if there is an $O(n^{2-\epsilon'})$ time algorithm for an $\epsilon' > 0$ for Verification of Pareto Sum.*

The proof can be found in the full version. Thus, for subquadratic reductions, we can restrict ourselves to the Verification of Pareto Sum problem, which essentially only checks the dominance condition.

Let us now consider the natural problem of computing the Pareto Sum.

▶ **Definition 33** (Pareto Sum). *Given sets $A, B \subseteq \mathbb{Z}^d$, compute a set $C \subseteq \mathbb{Z}$, such that A, B, C satisfy the Inclusion, Dominance and Minimality condition.*

In the following, we argue why the lower bounds to Verification of Pareto Sum translate to lower bounds to Computation of the Pareto Sum. Formally, we prove:

▶ **Lemma 34.** *If there is an algorithm to compute the Pareto Sum C of sets $A, B \subseteq \mathbb{Z}^d$ in time $O(n^{2-\epsilon})$ for an $\epsilon > 0$ even when $C = \Theta(n)$, then one can also decide Verification of Pareto Sum of sets A, B, C in time $O(n^{2-\epsilon'})$ for an $\epsilon' > 0$.*

We conclude this section with our resulting hardness results for computing Pareto Sums.

▶ **Theorem 7** (Pareto Sum Computation Lower Bound). *The following conditional lower bounds hold for output-sensitive Pareto sum computation:*

1. *If there is $\epsilon > 0$ such that we can compute the Pareto sum C of $A, B \subseteq \mathbb{Z}^2$, whenever C is of size $\Theta(n)$, in time $O(n^{2-\epsilon})$, then the 3-SUM hypothesis fails (thus, for any $\mathsf{FOP}^k_{\mathbb{Z}}$ formula ϕ of inequality dimension at most 3, there is $\epsilon' > 0$ such that ϕ can be decided in time $O(n^{k-1-\epsilon'})$).*
2. *If for all $d \geq 2$, there is $\epsilon > 0$ such that we can compute the Pareto sum C of $A, B \subseteq \mathbb{Z}^d$, whenever C is of size $\Theta(n)$, in time $O(n^{2-\epsilon})$, then there is some $\epsilon' > 0$ such that we can decide all $\mathsf{FOP}_{\mathbb{Z}}$ formulas with k quantifiers not ending in $\exists\forall\exists$ or $\forall\exists\forall$ in time $O(n^{k-1-\epsilon'})$.*

8 Future Work

While we exhibit a pair of problems that is complete for the class $\mathsf{FOP}_{\mathbb{Z}}$, one could still ask whether there is a subquadratic reduction from Hausdorff distance under n Translations to Verification of Pareto Sum. As a result there would be a single complete problem (or rather the canonical multidimensional family of a single geometric problem) for $\mathsf{FOP}_{\mathbb{Z}}$.

Is Verification of Pareto Sum complete for the class $\mathsf{FOP}_{\mathbb{Z}}$?

Interestingly, previous completeness theorems [36] were able to establish a problem of quantifier structure $\forall\forall\exists$ (the Orthogonal Vectors problem) as complete by making use of a technique in [51] that was originally used to show subcubic equivalence between All-Pairs Negative Triangle and Negative Triangle. However, a major problem we encounter is that while the third quantifier in the Orthogonal Vectors problem ranges over a sparse (intuitively: subpolynomially sized) domain (i.e., the dimensions of the vectors), the third quantifier in Pareto Sum Verification ranges over a linearly sized domain (i.e., the set C).

Finally, we ask if our 3-SUM completeness result for arbitrary quantifier structures can be improved upon.

Can we establish a $d > 3$ such that 3-SUM is complete for $\mathsf{FOP}_{\mathbb{Z}}$ formulas of inequality dimension at most d?

References

1 Amir Abboud, Arturs Backurs, Karl Bringmann, and Marvin Künnemann. Fine-grained complexity of analyzing compressed data: Quantifying improvements over decompress-and-solve. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 192–203. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.26.

2 Amir Abboud, Arturs Backurs, Karl Bringmann, and Marvin Künnemann. Impossibility results for grammar-compressed linear algebra. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/645e6bfdd05d1a69c5e47b20f0a91d46-Abstract.html.

3 Amir Abboud, Karl Bringmann, Seri Khoury, and Or Zamir. Hardness of approximation in p via short cycle removal: cycle detection, distance oracles, and beyond. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1487–1500. ACM, 2022. doi:10.1145/3519935.3520066.

4 Amir Abboud and Kevin Lewi. Exact weight subgraphs and the k-sum conjecture. In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, volume 7965 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2013. doi:10.1007/978-3-642-39206-1_1.

5 Amir Abboud, Kevin Lewi, and Ryan Williams. Losing weight by gaining edges. In Andreas S. Schulz and Dorothea Wagner, editors, *Algorithms - ESA 2014 - 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings*, volume 8737 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2014. doi:10.1007/978-3-662-44777-2_1.

6 Amir Abboud and Virginia Vassilevska Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 434–443. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.53.

7 Haozhe An, Mohit Gurumukhani, Russell Impagliazzo, Michael Jaber, Marvin Künnemann, and Maria Paula Parga Nina. The fine-grained complexity of multi-dimensional ordering properties. *Algorithmica*, 84(11):3156–3191, 2022. doi:10.1007/S00453-022-01014-X.

8 Christian Artigues, Marie-José Huguet, Fallou Gueye, Frédéric Schettini, and Laurent Dezou. State-based accelerations and bidirectional search for bi-objective multi-modal shortest paths. *Transportation Research Part C: Emerging Technologies*, 27:233–259, 2013.

9 Arturs Backurs, Piotr Indyk, and Ludwig Schmidt. Better approximations for tree sparsity in nearly-linear time. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2215–2229. SIAM, 2017. doi:10.1137/1.9781611974782.145.

10 Stephen A Bloch, Jonathan F Buss, and Judy Goldsmith. How hard are n 2-hard problems? *ACM SIGACT News*, 25(2):83–85, 1994. doi:10.1145/181462.181465.

11 Jean-Daniel Boissonnat, Micha Sharir, Boaz Tagansky, and Mariette Yvinec. Voronoi diagrams in higher dimensions under certain polyhedral distance functions. *Discret. Comput. Geom.*, 19(4):485–519, 1998. doi:10.1007/PL00009366.

12 Karl Bringmann, Alejandro Cassis, Nick Fischer, and Marvin Künnemann. Fine-grained completeness for optimization in P. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPIcs*, pages 9:1–9:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.APPROX/RANDOM.2021.9.

13 Karl Bringmann, Alejandro Cassis, Nick Fischer, and Marvin Künnemann. A structural investigation of the approximability of polynomial-time problems. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 30:1–30:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ICALP.2022.30.

14 Karl Bringmann, Nick Fischer, and Marvin Künnemann. A fine-grained analogue of schaefer's theorem in P: dichotomy of exists^k-forall-quantified first-order graph properties. In Amir Shpilka, editor, *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, volume 137 of *LIPIcs*, pages 31:1–31:27. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICS.CCC.2019.31.

15 Karl Bringmann, Nick Fischer, and Vasileios Nakos. Sparse nonnegative convolution is equivalent to dense nonnegative convolution. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1711–1724. ACM, 2021. doi:10.1145/3406325.3451090.

16 Karl Bringmann, Nick Fischer, and Vasileios Nakos. Deterministic and las vegas algorithms for sparse nonnegative convolution. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 3069–3090. SIAM, 2022. doi:10.1137/1.9781611977073.119.

17 Karl Bringmann and Vasileios Nakos. Fast n-fold boolean convolution via additive combinatorics. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 41:1–41:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.ICALP.2021.41.

18 Karl Bringmann and André Nusser. Translating hausdorff is hard: Fine-grained lower bounds for hausdorff distance under translation. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry, SoCG 2021, June 7-11, 2021, Buffalo, NY, USA (Virtual Conference)*, volume 189 of *LIPIcs*, pages 18:1–18:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.SOCG.2021.18.

19 Timothy M. Chan. Minimum l_∞ hausdorff distance of point sets under translation: Generalizing klee's measure problem. In Erin W. Chambers and Joachim Gudmundsson, editors, *39th International Symposium on Computational Geometry, SoCG 2023, June 12-15, 2023, Dallas, Texas, USA*, volume 258 of *LIPIcs*, pages 24:1–24:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.SOCG.2023.24.

20 Timothy M. Chan and Moshe Lewenstein. Clustered integer 3sum via additive combinatorics. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 31–40. ACM, 2015. doi:10.1145/2746539.2746568.

21 Timothy M. Chan, Virginia Vassilevska Williams, and Yinzhan Xu. Hardness for triangle problems under even more believable hypotheses: reductions from real apsp, real 3sum, and OV. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1501–1514. ACM, 2022. doi:10.1145/3519935.3520032.

22 Timothy M. Chan, Virginia Vassilevska Williams, and Yinzhan Xu. Fredman's trick meets dominance product: Fine-grained complexity of unweighted apsp, 3sum counting, and more. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 419–432. ACM, 2023. doi:10.1145/3564246.3585237.

23 L. Paul Chew, Dorit Dor, Alon Efrat, and Klara Kedem. Geometric pattern matching in d -dimensional space. *Discret. Comput. Geom.*, 21(2):257–274, 1999. doi:10.1007/PL00009420.

24 L Paul Chew and Klara Kedem. Improvements on geometric pattern matching problems. In *Algorithm Theory—SWAT'92: Third Scandinavian Workshop on Algorithm Theory Helsinki, Finland, July 8–10, 1992 Proceedings 3*, pages 318–325. Springer, 1992. doi:10.1007/3-540-55706-7_28.

25 L. Paul Chew and Klara Kedem. Improvements on geometric pattern matching problems. In Otto Nurmi and Esko Ukkonen, editors, *Algorithm Theory - SWAT '92, Third Scandinavian Workshop on Algorithm Theory, Helsinki, Finland, July 8-10, 1992, Proceedings*, volume 621 of *Lecture Notes in Computer Science*, pages 318–325. Springer, 1992. doi:10.1007/3-540-55706-7_28.

26 Richard Cole and Ramesh Hariharan. Verifying candidate matches in sparse and wildcard matching. In John H. Reif, editor, *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 592–601. ACM, 2002. doi:10.1145/509907.509992.

27 Marek Cygan, Marcin Mucha, Karol Wegrzycki, and Michal Wlodarczyk. On problems equivalent to (min, +)-convolution. *ACM Trans. Algorithms*, 15(1):14:1–14:25, 2019. doi:10.1145/3293465.

28 Holger Dell, Marc Roth, and Philip Wellnitz. Counting answers to existential questions. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 113:1–113:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICS.ICALP.2019.113.

29 Bartlomiej Dudek, Pawel Gawrychowski, and Tatiana Starikovskaya. All non-trivial variants of 3-ldt are equivalent. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 974–981. ACM, 2020. doi:10.1145/3357713.3384275.

30 Matthias Ehrgott and Xavier Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR-spektrum*, 22:425–460, 2000. doi:10.1007/S002910000046.

31 Jeff Erickson. New lower bounds for convex hull problems in odd dimensions. *SIAM J. Comput.*, 28(4):1198–1214, 1999. doi:10.1137/S0097539797315410.

32 Nick Fischer, Marvin Künnemann, and Mirza Redzic. The effect of sparsity on k-dominating set and related first-order graph properties. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 4704–4727. SIAM, 2024. doi:10.1137/1.9781611977912.168.

33 Daniel Funke, Demian Hespe, Peter Sanders, Sabine Storandt, and Carina Truschel. Pareto sums of pareto sets: Lower bounds and algorithms. *CoRR*, abs/2409.10232, 2024. doi:10.48550/arXiv.2409.10232.

34 Harold N. Gabow, Jon Louis Bentley, and Robert Endre Tarjan. Scaling and related techniques for geometry problems. In Richard A. DeMillo, editor, *Proceedings of the 16th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1984, Washington, DC, USA*, pages 135–143. ACM, 1984. doi:10.1145/800057.808675.

35 Anka Gajentaan and Mark H. Overmars. On a class of o(n2) problems in computational geometry. *Comput. Geom.*, 5:165–185, 1995. doi:10.1016/0925-7721(95)00022-2.

36 Jiawei Gao, Russell Impagliazzo, Antonina Kolokolova, and Ryan Williams. Completeness for first-order properties on sparse structures with algorithmic applications. *ACM Trans. Algorithms*, 15(2):23:1–23:35, 2019. doi:10.1145/3196275.

37 Demian Hespe, Peter Sanders, Sabine Storandt, and Carina Truschel. Pareto sums of pareto sets. In Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman, editors, *31st Annual European Symposium on Algorithms, ESA 2023, September 4-6, 2023, Amsterdam, The Netherlands*, volume 274 of *LIPIcs*, pages 60:1–60:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.ESA.2023.60.

38 Daniel P. Huttenlocher and Klara Kedem. Computing the minimum hausdorff distance for point sets under translation. In Raimund Seidel, editor, *Proceedings of the Sixth Annual Symposium on Computational Geometry, Berkeley, CA, USA, June 6-8, 1990*, pages 340–349. ACM, 1990. doi:10.1145/98524.98599.

39 Zahra Jafargholi and Emanuele Viola. 3sum, 3xor, triangles. *Algorithmica*, 74(1):326–343, 2016. doi:10.1007/S00453-014-9946-9.

40 Tsvi Kopelowitz, Seth Pettie, and Ely Porat. Higher lower bounds from the 3sum conjecture. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1272–1287. SIAM, 2016. doi:10.1137/1.9781611974331.CH89.

41 Marvin Künnemann. A tight (non-combinatorial) conditional lower bound for klee's measure problem in 3d. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 555–566. IEEE, 2022. doi:10.1109/FOCS54457.2022.00059.

42 Andrea Lincoln, Virginia Vassilevska Williams, Joshua R. Wang, and R. Ryan Williams. Deterministic time-space trade-offs for k-sum. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 58:1–58:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.ICALP.2016.58.

43 Andrea Lincoln, Virginia Vassilevska Williams, and R. Ryan Williams. Tight hardness for shortest cycles and paths in sparse graphs. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1236–1252. SIAM, 2018. doi:10.1137/1.9781611975031.80.

44 Thibaut Lust and Daniel Tuyttens. Variable and large neighborhood search to solve the multiobjective set covering problem. *Journal of Heuristics*, 20:165–188, 2014. doi:10.1007/S10732-013-9236-8.

45 André Nusser. *Fine-grained complexity and algorithm engineering of geometric similarity measures*. PhD thesis, Saarland University, Saarbrücken, Germany, 2021. URL: https://publikationen.sulb.uni-saarland.de/handle/20.500.11880/33904.

46 Mihai Puatracscu. Towards polynomial lower bounds for dynamic problems. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 603–610. ACM, 2010. doi:10.1145/1806689.1806772.

47 Britta Schulze, Kathrin Klamroth, and Michael Stiglmayr. Multi-objective unconstrained combinatorial optimization: a polynomial bound on the number of extreme supported solutions. *Journal of Global Optimization*, 74(3):495–522, 2019. doi:10.1007/S10898-019-00745-6.

48 Carola Wenk. *Shape matching in higher dimensions*. PhD thesis, Free University of Berlin, Dahlem, Germany, 2003. URL: http://www.diss.fu-berlin.de/2003/151/index.html.

49 Ryan Williams. Faster decision of first-order graph properties. In Thomas A. Henzinger and Dale Miller, editors, *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 80:1–80:6. ACM, 2014. doi:10.1145/2603088.2603121.

50 Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the international congress of mathematicians: Rio de janeiro 2018*, pages 3447–3487. World Scientific, 2018.

51 Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 645–654. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.67.

52 Virginia Vassilevska Williams and Ryan Williams. Finding, minimizing, and counting weighted subgraphs. *SIAM J. Comput.*, 42(3):831–854, 2013. doi:10.1137/09076619X.

Incompressible Functional Encryption

Rishab Goyal
University of Wisconsin-Madison, WI, USA

Venkata Koppula
IIT Delhi, India

Mahesh Sreekumar Rajasree[1]
IIT Delhi, India

Aman Verma
IIT Delhi, India

Abstract

Incompressible encryption (Dziembowski, Crypto'06; Guan, Wichs, Zhandry, Eurocrypt'22) protects from attackers that learn the entire decryption key, but cannot store the full ciphertext. In incompressible encryption, the attacker must try to compress a ciphertext within pre-specified memory bound S before receiving the secret key.

In this work, we generalize the notion of incompressibility to functional encryption. In incompressible functional encryption, the adversary can corrupt non-distinguishing keys at any point, but receives the distinguishing keys only after compressing the ciphertext to within S bits. An important efficiency measure for incompressible encryption is the ciphertext-rate (i.e., $\mathsf{rate} = |m|/|\mathsf{ct}|$). We give many new results for incompressible functional encryption for circuits, from minimal assumption of (non-incompressible) functional encryption, with

1. ct-rate-$\frac{1}{2}$ and *short* secret keys,
2. ct-rate-1 and *large* secret keys.

Along the way, we also give a new incompressible attribute-based encryption for circuits from standard assumptions, with ct-rate-$\frac{1}{2}$ and short secret keys. Our results achieve optimal efficiency, as incompressible attribute-based/functional encryption with ct-rate-1 as well as short secret keys has strong barriers for provable security from standard assumptions. Moreover, our assumptions are minimal as incompressible attribute-based/functional encryption are strictly stronger than their non-incompressible counterparts.

2012 ACM Subject Classification Security and privacy → Public key encryption

Keywords and phrases functional encryption, attribute-based encryption, incompressible encryption

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.56

Related Version *Full Version*: https://eprint.iacr.org/2024/798.pdf

Funding *Rishab Goyal*: Support for this research was provided by OVCRGE at UW–Madison with funding from the Wisconsin Alumni Research Foundation.

1 Introduction

A fundamental principle in cryptography is to leverage "secrets" for differentiating between an honest user legitimately accessing a system and an attacker trying to illegitimately access it. Consider data encryption as an example, where the goal is data secrecy. A central assumption underlying traditional encryption is that *an attacker cannot learn anything about a user's secret decryption key*. Moreover, this assumption seems inherent as if an attacker learns the secret key, then how can one distinguish between "capabilities" of an honest user and an attacker!?

[1] Corresponding author

© Rishab Goyal, Venkata Koppula, Mahesh Sreekumar Rajasree, and Aman Verma;
licensed under Creative Commons License CC-BY 4.0

In 2006, Dziembowzki [14] proposed an exquisite approach to *restrict the "capabilities" of an attacker*, even when the *entire* secret key gets corrupted. The proposal was to view *long-term memory storage* as a limited and expensive resource. In words, the intuition was that no real-world attacker can maintain an unbounded long-term memory, and it has to pick and choose what it wants to store in its memory. Dziembowzki [14] formalized this in the secret-key setting, and defined it as a *forward-secure storage* system. Recently, Guan, Wichs, and Zhandry (henceforth GWZ) [23] generalized this to the public-key setting. They labeled cryptography in this model as "incompressible" cryptography. The intuition being that an attacker cannot "compress" its unbounded short-term memory into its bounded long-term memory. GWZ provided multiple constructions for "incompressible" public-key encryption and signatures, from a variety of cryptographic assumptions with different efficiency tradeoffs.

Beyond Public-Key Encryption

Functional encryption (FE) [32, 2, 28, 7] is a powerful generalization of public-key encryption (PKE) [12]. FE generalizes PKE by providing fine-grained access to encrypted data. In FE, the master decryption key holder can create partial "functional" decryption keys sk_f for any supported function f of its choice. Such a key holder can recover $f(m)$ from any ciphertext ct, encrypting data m. Standard indistinguishability-based FE security states that no polytime attacker cannot distinguish between encryptions $\mathsf{ct}_0, \mathsf{ct}_1$ of messages m_0, m_1, unless it receives a secret key sk_{f^*} s.t. $f^*(m_0) \neq f^*(m_1)$. That is, unless an attacker receives a "distinguishing key" sk_{f^*}, it cannot distinguish between any two ciphertexts.[2] The ability to learn only function evaluation of encrypted data and nothing more has had fascinating consequences in cryptography, both theoretically and practically [33, 5, 11, 32, 22, 8, 2, 28, 7, 30, 17, 15, 20, 34, 10, 19, 29, 31, 21, 26, 27, 4, 13].

While functional keys offer fine-grained access over encrypted data, enabling many superior applications, they also enable far more attack opportunities compared to plain encryption. In PKE, there is just one secret key, and while that implies an all-or-nothing functionality, storing the key securely for the entire system lifetime is much easier. On the contrary, in FE, there are master secret keys as well as functional decryption keys. Each decryption key must be securely generated, distributed, and stored by the appropriate user. While it might be reasonable to expect the central trusted authority will (in most part) store master key msk securely; expecting this from every other user (holding a functional key) is unrealistic.

Our Contributions: Incompressible FE

In this work, we study incompressible functional encryption systems. We formalize the concept, and provide multiple constructions from a variety of cryptographic assumptions enabling different efficiency tradeoffs. As we elaborate in the full version [18], incompressibility in FE is extremely interesting, both theoretically and practically. In short, it pushes the traditional perception in FE that leaking an entire decryption key is equivalent to learning the function output on every previously encrypted data.

Formalizing incompressible FE. Interestingly, formally defining incompressibility for FE is more nuanced than for PKE. In public-key encryption, there is just one secret key. Thus, to define incompressibility for PKE, one considers a simple two-stage attacker: in stage 1,

[2] Throughout the sequel, by the phrase a "distinguishing key", we mean a secret key that can distinguish between two ciphertexts (by running decryption). Similarly, by the phrase "non-distinguishing keys", we mean secret keys that do not enable a trivial distinguishing attack between two ciphertexts.

the attacker receives a ciphertext that it must "compress" down to S bits[3]; and stage 2, where the attacker receives the secret key (along with the S-bit "compressed" ciphertext). As long as such an attacker cannot learn anything about the message, the PKE scheme is a secure incompressible scheme. In functional encryption setting, there are many more secret keys! Thus, depending upon the secret key(s) that a *stage 2* attacker learns, there are different formulations of incompressible FE security. Below, we briefly summarize three such formulations, and discuss them in further detail in the full version [18].[4]

- Standard security: the adversary receives *one* (or bounded number of) *distinguishing* secret key(s), after compressing challenge ciphertext (i.e., in stage 2). Moreover, it receives unbounded number of *non-distinguishing* secret keys before and after compressing the ciphertext.
- Semi-strong security: this is a strengthening of standard security, where the adversary can get an *unbounded* number of distinguishing keys in stage 2.
- Strong security: this further strengthens as now adversary gets the *master secret key* in stage 2.

Measuring incompressibility efficiency. Clearly, for the above to not be trivially impossible, the ciphertext size should be larger than S, the incompressibility limit. Further, the ciphertext should be larger than $|m|$ to uniquely encode m and guarantee correctness. However, it is not clear if there are any other necessary constraints on parameter sizes, beyond $|\mathsf{ct}| = \max(S, |m|) + \mathsf{poly}(\lambda)$. Here the $\max(S, |m|)$ term ensures both ciphertext constraints (i.e., $|\mathsf{ct}| > S$ and $|\mathsf{ct}| > |m|$).

Thus, an ideal goal is to design incompressible FE where $|\mathsf{mpk}|, |\mathsf{msk}|, |\mathsf{sk}_f|$ are all independent of S, and $|\mathsf{ct}| = \max(S, |m|) + \mathsf{poly}(\lambda)$. Such an incompressible FE scheme would have asymptotically optimal-sized parameters. As we discuss in the full version [18], achieving optimality in all parameter sizes is unachievable from standard falsifiable assumptions. Therefore, a standard approach in the literature [23, 9] is to design encryption with highest ciphertext-rate. Ciphertext-rate is defined as the ratio of message length and ciphertext length, i.e. $\frac{|m|}{|\mathsf{ct}|}$.

Prior works [23, 9] built incompressible PKE with optimal ct-rate of 1. Unfortunately, that comes at the cost of large secret keys, i.e. $|\mathsf{sk}| = \mathsf{poly}(S, \lambda)$. In this work, we design incompressible ABE/FE in two incomparable parameter regimes – ct-rate of 1 with *"large"* secret keys, and ct-rate of $\frac{1}{2}$ with *"short"* secret keys. We believe both parameter regimes to be equally interesting. In FE applications, where an honest user stores only secret keys and not ciphertexts in its persistent long-term storage, it might be beneficial to have short keys with constant ct-rate. While, in applications where an honest user stores only a few secret keys and a large number of ciphertexts, having a ct-rate of 1 might be more useful.

Our results

We summarize our main results below. All our schemes are in the standard model and under standard assumptions. We do not make any ideal-cipher, random oracle, or any other non-standard/non-falsifiable cryptographic assumption.

[3] S is typically considered to be the bound on the adversary's long-term storage.

[4] We can also consider joint compression of ciphertexts and secret keys in FE. The focus of this work is only on ciphertext incompressibility as discussed in the full version [18]. It is an interesting open question of whether new tradeoffs and applications can be enabled, if one considers joint incompressibility of ciphertexts and secret keys.

Incompressible FE. Our first set of results contains new constructions for incompressible FE for polynomial-sized circuits. We provide three separate and incomparable FE constructions as each provides a unique efficiency-security tradeoff. All three construction rely on a standard public-key FE scheme for polynomial-sized circuits.

1. Our first incompressible FE scheme is proven secure in the *semi-strong* incompressibility model. If the underlying FE scheme has ct-rate of $r \in [0, 1]$, then our incompressible FE scheme has an $\frac{r}{2}$ ct-rate.
2. Our second scheme is also proven secure in the *semi-strong* incompressibility model, except we only prove selective security.[5] Moreover, this construction is a ct-rate preserving construction. That is, we obtain ct-rate r incompressible FE scheme from any ct-rate r standard FE scheme. However, the secret keys are large to avoid known barriers.
3. Our third scheme is an extension of our above scheme. It is proven secure in the *selective standard* incompressibility model, but it provides a rather non-trivial efficiency guarantee. We prove that if the underlying FE scheme has short keys, then so does our incompressible FE scheme (without lowering the ct-rate). As we elaborate later in the overview, this does not contradict known implausibility results (see discussion in the full version [18]), but rather tightly matches it.

By plugging in known optimal-ciphertext-rate FE schemes [23, 25], we obtain incompressible FE schemes under the assumption of poly-secure FE with:

1. a ct-rate-$\frac{1}{2}$, short keys, and selective semi-strong security,
2. a ct-rate-$\frac{1}{4}$, short keys, and *adaptive* semi-strong security,
3. a ct-rate-1, large keys, and selective semi-strong security,
4. a ct-rate-1, *short* keys, and selective *standard* security.

Incompressible ABE. Our second result is an incompressible attribute-based encryption (ABE) scheme for polynomial-sized circuits achieves (standard) incompressible ABE security, with ciphertext size $|\mathsf{ct}| = S + |m| + \mathsf{poly}(\lambda)$ and all other parameter sizes, $|\mathsf{mpk}|, |\mathsf{msk}|, |\mathsf{sk}_f|$, are independent of S. That is, this scheme has ct-rate of $\frac{1}{2}$, and short keys. The construction and security proof are available in the full version [18].

2 Preliminaries and Notations

Throughout this paper, we will use λ to denote the security parameter and $\mathrm{negl}(\cdot)$ to denote a negligible function in the input. We will use the short-hand notation PPT for "probabilistic polynomial time". For any finite set X, $x \leftarrow X$ denotes the process of picking an element x from X uniformly at random. Similarly, for any distribution $\mathcal{D}$, $x \leftarrow \mathcal{D}$ denotes an element x drawn from the distribution $\mathcal{D}$. For any natural number $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, 2, \ldots, n\}$. For any two binary string x and y, $x||y$ denotes the concatenation of x and y. We use the following two notations to denote a family of circuits - $\{\mathcal{C}_n\}_n$ is a set of families of circuits indexed by some parameter n and $\{\mathcal{C}_{d,\ell}\}_{d,\ell}$ is a set of families of circuits indexed by the depth of the circuits d and the number of inputs to the circuits ℓ.

[5] Here selective means that the adversary must submit challenge messages before receiving secret keys in stage 1.

2.1 Randomness Extractors

▶ **Definition 1** (Strong Average Min-Entropy Extractor). *A (k, ϵ)-strong average min-entropy extractor is an efficient function* $\mathsf{Ext} : \{0,1\}^d \times \{0,1\}^n \to \{0,1\}^m$ *such that for all jointly distributed random variable X, Y where X takes values $\{0,1\}^d$ and $H_\infty(X|Y) \geq k$, we have $(U_d, \mathsf{Ext}(X, U_d), Y) \approx_\epsilon (U_d, U_m, Y)$ where U_d, U_m are uniformly random strings of length d, m respectively. Here $H_\infty(X|Y) = -\log \mathop{\mathbb{E}}_{y \leftarrow Y}(\max_x \mathsf{Pr}(X = x|Y = y))$ is the average min-entropy of X conditioned on Y.*

▶ **Theorem 2.** *There exists an explicit efficient $(k, 2^{-\lambda})$-strong average min-entropy extractor* $\mathsf{Ext} : \{0,1\}^d \times \{0,1\}^n \to \{0,1\}^\lambda$ *such that $k = \mathsf{poly}(\lambda)$, $d = \mathsf{poly}(\lambda)$, $n = S + k$ and the depth of the extractor circuit is* $\mathsf{poly}(\lambda, \log(n))$.

2.2 Pseudorandom Functions

A family of pseudorandom functions $\mathsf{PRF} = (\mathsf{KeyGen}, \mathsf{Eval})$ with key space $\{\mathcal{K}_\lambda\}_\lambda$, input space $\{\mathcal{X}_\lambda\}_\lambda$ and output space $\{\mathcal{Y}_\lambda\}_\lambda$ consists of the following algorithms.

- $\mathsf{KeyGen}(1^\lambda)$: The key generation algorithm is a randomized algorithm that takes as input the security parameter 1^λ and outputs a key $k \in \mathcal{K}_\lambda$.
- $\mathsf{Eval}(k, x)$: The evaluation algorithm is a deterministic algorithm that takes as input a key $k \in \mathcal{K}_\lambda$ and $x \in \mathcal{X}_\lambda$ and outputs $y \in \mathcal{Y}_\lambda$.

▶ **Definition 3.** *A PRF scheme* PRF *is secure if for all PPT adversary $\mathcal{A}$, there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that for all $\lambda \in \mathbb{N}$,*

$$|\Pr[\mathcal{A}^{\mathsf{Eval}(k,\cdot)}(1^\lambda) = 1 \ : \ k \leftarrow \mathsf{KeyGen}(1^\lambda)] - \Pr[\mathcal{A}^{R(\cdot)}(1^\lambda) = 1 \ : \ R \leftarrow \mathcal{U}_\lambda]| \leq \mathrm{negl}(\lambda)$$

where $\mathcal{U}_\lambda$ is the set of all functions from $\mathcal{X}_\lambda$ to $\mathcal{Y}_\lambda$.

2.3 Garbling Scheme

A garbling scheme $\mathsf{GC} = (\mathsf{Garble}, \mathsf{Eval})$ for a class of circuits $\{\mathcal{C}_\lambda\}_\lambda$ consists of the following algorithms.

- $\mathsf{Garble}(1^\lambda, C)$: The garbling algorithm is a randomized algorithm that takes as input the security parameter 1^λ and a circuit $C \in \mathcal{C}_\lambda$ such that $C : \{0,1\}^n \to \{0,1\}$ and outputs a garbled circuit $\hat{C}$ and a set of labels $\{\mathsf{lab}_{i,b}\}_{i \in [n], b \in \{0,1\}}$.
- $\mathsf{Eval}(\hat{C}, \{\mathsf{lab}_i\}_{i \in [n]})$: The evaluation algorithm takes as input a garbled circuit $\hat{C}$ and a set of n labels $\{\mathsf{lab}_i\}_{i \in [n]}$ and outputs $y \in \{0,1\}$.

Correctness

For correctness of a garbling scheme GC for a class of circuits $\{\mathcal{C}_\lambda\}_\lambda$, we require that for all $\lambda \in \mathbb{N}, C \in \mathcal{C}_\lambda$, $x \in \{0,1\}^n$,

$$\mathsf{Eval}(\hat{C}, \{\mathsf{lab}_{i,x_i}\}_{i \in [n]}) = C(x)$$

where $(\hat{C}, \{\mathsf{lab}_{i,b}\}_{i \in [n], b \in \{0,1\}}) \leftarrow \mathsf{Garble}(1^\lambda, C)$.

Security

For security, we define simulation based security [35, 1].

▶ **Definition 4.** *A garbling scheme* $\mathsf{GC} = (\mathsf{Garble}, \mathsf{Eval})$ *for a class of circuits* $\{\mathcal{C}_\lambda\}_\lambda$ *is said to be secure if there exists a PPT algorithm* Sim *such that for all PPT adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that for all* $\lambda \in \mathbb{N}$*, the following holds.*

$$\left| \Pr\left[\mathcal{A}_2(\hat{C}, \{\mathsf{lab}_{i,x_i}\}_{i\in[n]}, \mathsf{aux}) = 1 : \begin{array}{c}(C, x, \mathsf{aux}) \leftarrow \mathcal{A}_1(1^\lambda),\\ (\hat{C}, \{\mathsf{lab}_{i,x_i}\}_{i\in[n]}) \leftarrow \mathsf{Sim}(1^\lambda, 1^n, 1^{|C|}, C(x))\end{array}\right] \right.$$

$$\left. - \Pr\left[\mathcal{A}_2(\hat{C}, \{\mathsf{lab}_{i,x_i}\}_{i\in[n]}, \mathsf{aux}) = 1 : \begin{array}{c}(C, x, \mathsf{aux}) \leftarrow \mathcal{A}_1(1^\lambda),\\ (\hat{C}, \{\mathsf{lab}_{i,b}\}_{i\in[n],b\in\{0,1\}}) \leftarrow \mathsf{Garble}(1^\lambda, C))\end{array}\right] \right| \leq \frac{1}{2} + \mathrm{negl}(\lambda).$$

2.4 Incompressible Secret Key Encryption

An incompressible secret key encryption scheme $\mathsf{IncSKE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\{\mathcal{M}_\lambda\}_\lambda$ consists of the following PPT algorithms.

- $\mathsf{Setup}(1^\lambda, 1^S, 1^n)$: The setup algorithm is a randomized algorithm that takes as input the security parameter λ, a parameter 1^S, the length of the message 1^n and outputs a secret key sk.
- $\mathsf{Enc}(\mathsf{sk}, m)$: The encryption algorithm is a randomized algorithm that takes as input a secret key sk and a message $m \in \mathcal{M}_\lambda$ and outputs a ciphertext ct.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$: The decryption algorithm takes as input a secret key sk and a ciphertext ct and outputs either a message $m \in \mathcal{M}_\lambda$ or $\perp$.

Correctness

For correctness, we require that for all $\lambda \in \mathbb{N}, S \in \mathbb{N}, n \in \mathbb{N}, m \in \mathcal{M}_\lambda$ and $\mathsf{sk} \leftarrow \mathsf{Setup}(1^\lambda, 1^S, 1^n)$,

$$\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{sk}, m)) = m] = 1$$

where the probability is over the random bits used in the encryption algorithm.

Incompressible SKE Security

Consider the following experiment with an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.

- **Initialization Phase:** $\mathcal{A}_1$ on input 1^λ, outputs an upper bound on the state size 1^S and the length of the message 1^n. The challenger runs $\mathsf{sk} \leftarrow \mathsf{Setup}(1^\lambda, 1^S, 1^n)$.
- **Challenge Phase:** $\mathcal{A}_1$ outputs a message m, along with an auxiliary information aux. The challenger randomly chooses $b \in \{0, 1\}$. If $b = 0$, it samples a truly random string ct^*. Else, it computes a ciphertext $\mathsf{ct}^* = \mathsf{Enc}(\mathsf{sk}, m)$ and sends it to $\mathcal{A}_1$.[6]
- **First Response Phase:** $\mathcal{A}_1$ computes a state st such that $|\mathsf{st}| \leq S$.
- **Second Response Phase:** $\mathcal{A}_2$ receives $(\mathsf{sk}, \mathsf{aux}, \mathsf{st})$ and outputs b'. $\mathcal{A}$ wins the experiment if $b = b'$.

▶ **Definition 5.** *An SKE scheme is said to be incompressible secure if for all PPT adversaries* $\mathcal{A}$*, there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that for all* $\lambda \in \mathbb{N}$*,*

$$\Pr[\mathcal{A} \textit{ wins in the above experiment}] \leq \frac{1}{2} + \mathrm{negl}(\lambda)$$

[6] In the incompressible security definition presented in prior works [9, 24], the adversary sends two messages m_0, m_1 and receives an encryption of one of these message. Note that this is a weaker notion, which is implied by the incompressible security notion defined in this paper.

CPA-SKE Security

Consider the following experiment with an adversary $\mathcal{A}$ where Setup algorithm takes only 1^λ and 1^n as input.

- **Initialization Phase:** The challenger runs $\mathsf{sk} \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$.
- **Pre-Challenge Query Phase:** $\mathcal{A}$ is allowed to make polynomially many queries. For each query m, the challenger computes $\mathsf{ct} \leftarrow \mathsf{SKE.Enc}(\mathsf{sk}, m)$ and returns ct to Adv.
- **Challenge Phase:** $\mathcal{A}$ outputs a message m^*. The challenger randomly chooses $b \in \{0,1\}$. If $b = 0$, it samples a truly random string ct^*. Else, it computes a ciphertext $\mathsf{ct}^* = \mathsf{Enc}(\mathsf{sk}, m^*)$ and sends it to $\mathcal{A}$.[7]
- **Post-Challenge Query Phase:** $\mathcal{A}$ is allowed to make polynomially many queries. For each query m, the challenger computes $\mathsf{ct} \leftarrow \mathsf{SKE.Enc}(\mathsf{sk}, m)$ and returns ct to Adv.
- **Response Phase:** $\mathcal{A}$ outputs $b' \in \{0,1\}$ and wins the experiment if $b = b'$.

▶ **Definition 6.** *An SKE scheme is said to be secure if for all PPT adversaries $\mathcal{A}$, there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that for all $\lambda \in \mathbb{N}, n \in \mathbb{N}$,*

$$\Pr[\mathcal{A} \text{ wins in the above experiment}] \leq \frac{1}{2} + \mathrm{negl}(\lambda)$$

▶ **Theorem 7** ([14])**.** *Assuming the existence of one-way functions, there exists an incompressible SKE scheme with* $\mathsf{poly}(\lambda)$ *secret-key size and $S + n + \mathsf{poly}(\lambda)$ ciphertext size where n is the size of the message and S is the compressibility parameter. The depth of the decryption circuit is* $\mathsf{poly}(\lambda, \log(S, n))$.

2.5 Functional Encryption

A functional encryption scheme $\mathsf{FE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ for the function space $\{\mathcal{F}_n : \mathcal{X}_n \to \mathcal{Y}_n\}_n$ consists of the following PPT algorithms.

- $\mathsf{Setup}(1^\lambda, 1^n)$: The setup algorithm is a randomized algorithm that takes as input the security parameter 1^λ and an index 1^n and outputs a master public key mpk and a secret key msk.
- $\mathsf{KeyGen}(\mathsf{msk}, f)$: The key generation algorithm is a randomized algorithm that takes as input the master secret msk and a function $f \in \mathcal{F}_n$ and outputs a secret sk_f.
- $\mathsf{Enc}(\mathsf{mpk}, m)$: The encryption algorithm is a randomized algorithm that takes as input a public key mpk and a message $m \in \mathcal{X}_n$ and outputs a ciphertext ct.
- $\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct})$: The decryption algorithm takes as input a secret key sk_f and a ciphertext ct and outputs either a $y \in \mathcal{Y}_n$ or $\perp$.

Correctness

For correctness, we require that for all $\lambda \in \mathbb{N}, n \in \mathbb{N}, m \in \mathcal{X}_n, f \in \mathcal{F}_n$ and $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$,

$$\Pr[\mathsf{Dec}(\mathsf{KeyGen}(\mathsf{msk}, f), \mathsf{Enc}(\mathsf{mpk}, m)) = f(m)] = 1$$

where the probability is over the random bits used in the encryption and key generation algorithm.

[7] Note that this security notion implies the standard indistinguishability security notion where the adversary sends two messages m_0, m_1 and receives an encryption of one of the messages.

Adaptive IND-based Security

Consider the following experiment with an adversary $\mathcal{A}$.

- **Initialization Phase:** The challenger runs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$ and sends mpk to $\mathcal{A}$.
- **Pre-Challenge Query Phase:** $\mathcal{A}$ is allowed to make polynomially many queries. For each query f, the challenger computes $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{msk}, f)$ and returns sk_f to $\mathcal{A}$.
- **Challenge Phase:** $\mathcal{A}$ outputs two message m_0, m_1. If there exists a function f queried by $\mathcal{A}$ such that $f(m_0) \neq f(m_1)$, the challenger aborts the game. Else, it randomly chooses $b \in \{0, 1\}$ and computes a ciphertext $\mathsf{ct}^* = \mathsf{Enc}(\mathsf{mpk}, m_b)$ and sends it to $\mathcal{A}$.
- **Post-Challenge Query Phase:** $\mathcal{A}$ is allowed to make polynomially many queries. For each query f, if $f(m_0) \neq f(m_1)$, the challenger sends $\perp$. Else, computes $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{msk}, f)$ and returns sk_f to $\mathcal{A}$.
- **Response Phase:** $\mathcal{A}$ outputs b'. $\mathcal{A}$ wins the experiment if $b = b'$.

▶ **Definition 8.** *An FE scheme satisfies adaptive indistinguishability-based security if for all PPT adversaries $\mathcal{A}$, there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that for all $\lambda \in \mathbb{N}$,*

$$\Pr[\mathcal{A} \textit{ wins in the above experiment}] \leq \frac{1}{2} + \mathrm{negl}(\lambda)$$

If the FE is secure against an unbounded number of queries from the adversary, then we say that the scheme is collision-resistant.

A relaxed notion of the FE indistinguishability-based security is the well-known selective security model. An FE scheme is said to be *selectively* secure if it is secure against PPT adversaries $\mathcal{A}$ that do not make any key queries during the pre-challenge query phase, and have to submit the challenge message before obtaining the public parameters.

The following two theorems addresses the optimal ciphertext-rate and secret-key size FE schemes for both adaptive and selective settings. It is important to note that in these FE scheme, the decryption algorithm requires the complete description of the function f in addition to the secret key sk_f to perform decryption.

▶ **Theorem 9** ([25])**.** *Assuming selectively secure FE for circuits, there exists an adaptively secure FE such that*

$$|\mathsf{mpk}| = O_\lambda(1), \quad |\mathsf{sk}_f| = O_\lambda(1)^8, \quad |\mathsf{ct}| = 2|x| + O_\lambda(1)$$

where $O_\lambda(\cdot)$ hides factors of $\mathsf{poly}(\lambda)$*, λ is the security parameter,* ct *is a ciphertext and* sk_f *is a secret key for the function f, x is any element from the input space of f and* mpk *is the master public key generated by the scheme.*

▶ **Theorem 10** ([25])**.** *Assuming selectively secure FE for circuits, there exists an adaptively secure FE such that*

$$|\mathsf{mpk}| = O_\lambda(1), \quad |\mathsf{sk}_f| = O_\lambda(1)^9, \quad |\mathsf{ct}| = |x| + O_\lambda(1)$$

where $O_\lambda(\cdot)$ hides factors of $\mathsf{poly}(\lambda)$*, λ is the security parameter,* ct *is a ciphertext and* sk_f *is a secret key for the function f that outputs a single bit, x is any element from the input space of f and* mpk *is the master public key generated by the scheme.*

Simulation security

Consider the following real and simulated experiment with an adversary $\mathcal{A}$ where Sim is a PPT simulator.

Real Experiment

- **Initialization Phase:** The challenger runs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$ and sends mpk to $\mathcal{A}$.
- **Challenge Phase:** $\mathcal{A}$ outputs a message m and a function f. The challenger computes a ciphertext $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, m)$ and $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{msk}, f)$ and sends $(\mathsf{ct}, \mathsf{sk}_f)$ to $\mathcal{A}$.
- **Response Phase:** $\mathcal{A}$ outputs b.

Simulated Experiment

- **Initialization Phase:** The simulator runs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$ and sends mpk to $\mathcal{A}$.
- **Challenge Phase:** $\mathcal{A}$ outputs a message m and a function f. The simulator computes $(\mathsf{ct}, \mathsf{sk}_f) \leftarrow \mathsf{Sim}(\mathsf{mpk}, f, f(m), 1^{|m|})$ and sends $(\mathsf{ct}, \mathsf{sk}_f)$ to $\mathcal{A}$.
- **Response Phase:** $\mathcal{A}$ outputs b.

▶ **Definition 11.** *An FE scheme is said to be* ***single-key*** *simulation secure if for all PPT adversaries $\mathcal{A}$, there exists a PPT simulator* Sim *and a negligible function* $\mathrm{negl}(\cdot)$ *such that for all $\lambda \in \mathbb{N}, n \in \mathbb{N}$,*

$$\Big|\Pr[\mathcal{A} \text{ outputs 0 in the real experiment}] - \Pr[\mathcal{A} \text{ outputs 0 in the simulated experiment}]\Big| \leq \mathrm{negl}(\lambda)$$

▶ **Theorem 12** ([16])**.** *Assuming the hardness of* LWE*, there exists a single-key simulation secure* FE *for a class of circuits $\{\mathcal{C}_{n,d}\}_{n,d}$ that takes input of size n, outputs a single bit and has depth d such that*

$$|\mathsf{ct}| = O_\lambda(n \cdot d^2), \quad |\mathsf{sk}_f| = O_\lambda(d^2), \quad |mpk| = O_\lambda(n \cdot d^2)$$

where $O_\lambda(\cdot)$ hides factors of $\mathsf{poly}(\lambda)$, λ is the security parameter, ct *is a ciphertext and* sk_f *is a secret key for the circuit $f \in \mathcal{C}_{n,d}$ and* mpk *is the master public key generated by the scheme.*

▶ **Corollary 13.** *Assuming the hardness of* LWE*, there exists a single-key simulation secure* FE *for a class of circuits $\{\mathcal{C}_{n,d}\}_{n,d}$ that takes input of size n, outputs m-bits and has depth d such that*

$$|\mathsf{ct}| = O_\lambda(m \cdot n \cdot d^2), \quad |\mathsf{sk}_f| = O_\lambda(m \cdot d^2), \quad |mpk| = O_\lambda(m \cdot n \cdot d^2)$$

where $O_\lambda(\cdot)$ hides factors of $\mathsf{poly}(\lambda)$, λ is the security parameter, ct *is a ciphertext and* sk_f *is a secret key for the circuit $f \in \mathcal{C}_{n,d}$ and* mpk *is the master public key generated by the scheme.*

Specializing FE

An attribute based encryption scheme (ABE) is a function encryption scheme where the message space is $\{\mathcal{M}_n \times \mathcal{X}_n\}_n$, i.e., it is a tuple of a message and an attribute and the function class is a set of function $\{\mathcal{F}_n\}_n$ such that any function $f_C \in \mathcal{F}_n$ is associated with a predicate function C with the following definition.

$$\mathcal{F}_C(m, x) = \begin{cases} m & if \;\; C(x) = 1 \\ \perp & if \;\; C(x) = 0 \end{cases}$$

In the security experiment, the adversary has to output m_0, m_1, x^* in the challenge phase with the restriction that it has never queried a key for a function f_C such that $C(x^*) = 1$.

▶ **Theorem 14** ([6]). *Assuming the hardness of* LWE*, there exists an selectively secure ABE scheme for any family of circuits* $\{\mathcal{C}_{d,\ell}\}_{d,\ell}$ *that has* ℓ*-length input and* d*-depth satisfying*

$$|\mathsf{ct}| = O_\lambda(\ell \cdot d^2), \qquad |\mathsf{sk}_f| = O_\lambda(d^2), \qquad |mpk| = O_\lambda(\ell \cdot d^2)$$

where $O_\lambda(\cdot)$ *hides factors of* $\mathsf{poly}(\lambda)$*,* λ *is the security parameter,* ct *is a ciphertext and* sk_f *is a secret key for the predicate* $f \in \mathcal{C}_{d,\ell}$ *and* mpk *is the master public key generated by the scheme.*

An identity based encryption scheme (IBE) is a attribute based encryption scheme where the message space is $\{\mathcal{M}_\lambda \times \mathcal{ID}_\lambda\}_\lambda$, i.e., it is a tuple of a message and an identity and the function class is a set of function $\{\mathcal{F}_\lambda\}_\lambda$ such that any function $f_{\mathsf{id}} \in \mathcal{F}_\lambda$ is associated with an identity id with the following definition.

$$\mathcal{F}_{\mathsf{id}}(m, \mathsf{id}') = \begin{cases} m & if \ \ \mathsf{id} = \mathsf{id}' \\ \perp & if \ \ \mathsf{id} \neq \mathsf{id}' \end{cases}$$

In the security experiment, the adversary has to output m_0, m_1, id^* in the challenge phase with the restriction that it has never queried a key for id^*.

3 Incompressible Functional Encryption: Definitions

In this section, we will proceed to define the incompressible version of the security game for FE scheme where Setup takes an additional input 1^S. Similar to the incompressible SKE schemes, we will consider two adversaries $\mathcal{A}_1, \mathcal{A}_2$. The first adversary $\mathcal{A}_1$ will be provided with the complete challenge ciphertext and produce a compressed version of it. The second adversary $\mathcal{A}_2$ is provided with the master public key, compressed challenge ciphertext which was created by $\mathcal{A}_1$ and certain secret keys.

▶ **Definition 15** (Incompressible FE Security). *Let* $\mathsf{FE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *be an FE scheme in which the setup algorithm takes an additional parameter* 1^S *as input. Consider the following experiment with an adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*.*

- ***Initialization Phase:*** *$\mathcal{A}_1$ on input 1^λ, outputs an upper bound on the state size 1^S and 1^n. The challenger runs $(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^S, 1^n)$ and sends mpk to $\mathcal{A}_1$.*
- ***Pre-Challenge Query Phase:*** *In this phase, $\mathcal{A}_1$ is allowed to make polynomially many key queries. For each query f sent to the challenger, the challenger computes $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{msk}, f)$ and returns sk_f to $\mathcal{A}_1$.*
- ***Challenge Phase:*** *$\mathcal{A}_1$ outputs two messages m_0, m_1, along with an auxiliary information aux. If there exists a function f queried by $\mathcal{A}_1$ such that $f(m_0) \neq f(m_1)$, the challenger aborts the game. Else, it randomly chooses $b \in \{0, 1\}$ and computes a ciphertext $\mathsf{ct}^* = \mathsf{Enc}(\mathsf{mpk}, m_b)$ and sends it to $\mathcal{A}_1$.*
- ***Post-Challenge Query Phase:*** *This is similar to the pre-challenge query phase. The adversary $\mathcal{A}_1$ is allowed to send polynomially many key queries. For each query f, if $f(m_0) \neq f(m_1)$, the challenger sends $\perp$. Else, computes $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{msk}, f)$ and returns sk_f to $\mathcal{A}_1$.*
- ***First Response Phase:*** *$\mathcal{A}_1$ computes a state st such that $|st| \leq S$.*
- ***Second Response Phase:*** *$\mathcal{A}_2$ receives (mpk, aux, st) and can make the following query.*
  - ***Key Query Phase:*** *In this phase, $\mathcal{A}_2$ is allowed to make **a single distinguishing key** query but polynomially many queries for non-distinguishing keys. The challenger computes $\mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{msk}, f)$ and returns sk_f to $\mathcal{A}_2$.*

  Finally, $\mathcal{A}_2$ outputs b'. $\mathcal{A}$ wins the experiment if $b = b'$.

An FE scheme is said to be incompressible secure if for all PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, there exists a negligible function* $\text{negl}(\cdot)$ *such that for all* $\lambda \in \mathbb{N}$*,*

$$\Pr[\mathcal{A} \text{ wins in the above experiment}] \leq \frac{1}{2} + \text{negl}(\lambda)$$

We consider two stronger variants of the above security - strongly incompressible and semi-strongly incompressible. In the strongly incompressible version, $\mathcal{A}_2$ is provided with the master secret msk whereas in the semi-strongly incompressible version, it is allowed to make polynomially many distinguishing key queries.

▶ **Definition 16** (Strongly Incompressible FE Security)**.** *An FE scheme is said to be strongly incompressible secure if for all PPT adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, there exists a negligible function* $\text{negl}(\cdot)$ *such that for all* $\lambda \in \mathbb{N}$*,*

$$\Pr[\mathcal{A} \text{ wins in the Incompressible } \mathsf{FE} \text{ experiment}] \leq \frac{1}{2} + \text{negl}(\lambda)$$

provided the second adversary $\mathcal{A}_2$ *is also given the* ***master secret key*** msk *at the beginning of Second Response Phase.*

▶ **Definition 17** (Semi-Strongly Incompressible FE Security)**.** *An FE scheme is said to be semi-strongly incompressible secure if for all PPT adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, there exists a negligible function* $\text{negl}(\cdot)$ *such that for all* $\lambda \in \mathbb{N}$*,*

$$\Pr[\mathcal{A} \text{ wins in the Incompressible } \mathsf{FE} \text{ experiment}] \leq \frac{1}{2} + \text{negl}(\lambda)$$

provided the second adversary $\mathcal{A}_2$ *is allowed to make* ***polynomially many distinguishing key queries*** *in the Key Query Phase.*

We can consider similar incompressible security notions for IBE and ABE schemes.

4 Rate-$\frac{1}{2}$ Incompressible FE with Short Keys and Semi-Strong Security

In this section, we present a *semi-strong incompressible* FE scheme, using a regular FE scheme along with an incompressible SKE and regular SKE scheme. If the underlying FE has ct-rate of $r \in [0, 1]$, then our incompressible FE scheme has an $\frac{r}{2}$ ct-rate.

Our Construction

Our FE scheme supports functions from the class $\{\mathcal{F}_n : \mathcal{N}_n \to \{0,1\}\}_n$ such that any function in this class has a $\mathsf{polylog}(|\mathcal{N}_n|)$ depth circuit using the following primitives. Throughout S denotes the (in)compressibility parameter.

- $\mathsf{IncSKE} = (\mathsf{IncSKE.Setup}, \mathsf{IncSKE.Enc}, \mathsf{IncSKE.Dec})$ be an incompressible SKE. We define $\alpha = \alpha(\lambda, S)$ as the length of the secret key of this scheme and $\beta = \beta(\lambda, S)$ as the length of the ciphertext for a message of length λ encrypted using this scheme.
- $\mathsf{SKE} = (\mathsf{SKE.Setup}, \mathsf{SKE.Enc}, \mathsf{SKE.Dec})$ be an SKE scheme with secret keys of length λ.
- $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.Keygen}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ be an FE scheme with message spaces $\{\tilde{\mathcal{N}}_n\}_n$ and function space $\{\tilde{\mathcal{F}}_n\}_n$ that contains the decryption circuits of IncSKE and SKE.

Let $\tilde{n} = \tilde{n}(\lambda, n, S)$ be the lexicographically smallest functionality index such that $\tilde{\mathcal{F}}_{\tilde{n}}$ contains $\mathcal{F}_n$, the decryption circuit of SKE and IncSKE. We describe the algorithms for our incompressible FE scheme below.

- $\mathsf{Setup}(1^\lambda, 1^S, 1^n)$: The setup algorithm samples a master public key and a master secret key of the FE scheme by computing $(\mathsf{fe.mpk}, \mathsf{fe.msk}) \leftarrow \mathsf{FE.Setup}(1^\lambda, 1^{\tilde{n}})$. It generates a secret key of the incompressible SKE scheme by computing $\mathsf{inc.sk} \leftarrow \mathsf{IncSKE.Setup}(1^\lambda, 1^S, 1^\lambda)$ and two secret keys of the SKE scheme by computing $\mathsf{ske.sk} \leftarrow \mathsf{SKE.Setup}(1^\lambda)$ and $\mathsf{ske.sk}' \leftarrow \mathsf{SKE.Setup}(1^\lambda)$. It sets $\mathsf{mpk} = \mathsf{fe.mpk}$ and $\mathsf{msk} = (\mathsf{fe.msk}, \mathsf{inc.sk}, \mathsf{ske.sk}, \mathsf{ske.sk}')$.
- $\mathsf{KeyGen}(\mathsf{msk}, f)$: Let $\mathsf{msk} = (\mathsf{fe.msk}, \mathsf{inc.sk}, \mathsf{ske.sk}, \mathsf{ske.sk}')$. The key generation algorithm first computes ciphertext $\mathsf{ske.ct} = \mathsf{SKE.Enc}(\mathsf{ske.sk}, (0, 0^\alpha))$ and $\mathsf{ske.ct}' = \mathsf{SKE.Enc}(\mathsf{ske.sk}', 0)$. Then, it generates a key of the FE scheme by computing $\mathsf{fe.sk}_f \leftarrow \mathsf{FE.Keygen}(\mathsf{fe.msk}, C_{f,\mathsf{ske.ct},\mathsf{ske.ct}'})$ where the function $C_{f,\mathsf{ske.ct},\mathsf{ske.ct}'}$ is described in Figure 1.

Input: Messages m, SKE key $\mathsf{ske.sk}$, incompressible ciphertext $\mathsf{inc.ct}$ and a flag b
Hardwired: Function f, two SKE ciphertexts $\mathsf{ske.ct}, \mathsf{ske.ct}'$
Output: y

1. Check if $b = 0$:
   – If yes, output $y = f(m)$.
2. Otherwise, compute $(flag, \mathsf{inc.sk}) \leftarrow \mathsf{SKE.Dec}(\mathsf{ske.sk}, \mathsf{ske.ct})$.
3. Check if $flag = 0$:
   – If yes, output $y = f(m)$.
4. Otherwise, compute $\mathsf{ske.sk}' \leftarrow \mathsf{IncSKE.Dec}(\mathsf{inc.sk}, \mathsf{inc.ct})$.
5. Output $y = \mathsf{SKE.Dec}(\mathsf{ske.sk}', \mathsf{ske.ct}')$.

Figure 1 Description of $C_{f,\mathsf{ske.ct},\mathsf{ske.ct}'}$.

- $\mathsf{Enc}(\mathsf{mpk}, m)$: Let $\mathsf{mpk} = \mathsf{fe.mpk}$. The encryption algorithm randomly samples $\mathsf{inc.ct} \leftarrow \{0,1\}^\beta$ and generates $\mathsf{fe.ct} \leftarrow \mathsf{FE.Enc}(\mathsf{fe.mpk}, (m, \perp, \mathsf{inc.ct}, 0))$[10]. It returns $\mathsf{fe.ct}$.
- $\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct})$: Let $\mathsf{sk}_f = \mathsf{fe.sk}_f$. The decryption algorithm decrypts the ciphertext by computing $m = \mathsf{FE.Dec}(\mathsf{fe.sk}_f, \mathsf{ct})$. It returns m.

Correctness

A ciphertext corresponding to a message m in the above scheme is an encryption of $(m, \perp, \mathsf{inc.ct}, 0)$ using the FE scheme, i.e., $\mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{fe.mpk}, (m, \perp, \mathsf{inc.ct}, 0))$ where $\mathsf{inc.ct}$ is a truly random string.

Consider that a decryptor possesses the secret key sk_f for some function f. The secret key sk_f is a key $\mathsf{fe.sk}_f$ of the underlying FE scheme for the circuit $C_{f,\mathsf{ske.ct},\mathsf{ske.ct}'}$ which takes as input in $(m, \mathsf{ske.sk}, \mathsf{inc.ct}, b)$ and outputs $f(m)$ if $b = 0$. The decryption algorithm simply computes the decryption algorithm of the underlying FE scheme on the ciphertext ct using $\mathsf{fe.sk}_f$ and returns the output. Since, b is set to 0 in the ciphertext ct, decrypting ct using $\mathsf{fe.sk}_f$ gives $f(m)$ due to the correctness of the underlying FE scheme.

Size of the master public key

The master public key of our scheme is $\mathsf{fe.mpk}$. Therefore, the size is $|\mathsf{fe.mpk}|$.

[10] By $\perp$, we mean it sets a null secret key.

Size of the ciphertext

Recall that the ciphertext is an FE encryption of $(m, \perp, \mathsf{inc.ct}, 0)$ where the third component has size β. From the definition of β, it is the length of an incompressible ciphertext which is intended encrypt an SKE secret key, therefore, $\beta = S + \mathsf{poly}(\lambda)$ using Dziembowski's construction (see Theorem 7). So, the size of $(m, \perp, \mathsf{inc.ct}, 0)$ is $|m| + S + \mathsf{poly}(\lambda)$. Assuming that FE has ct-rate of r, the size of $\mathsf{fe.ct}$ is also $r^{-1} \cdot (|m| + S + \mathsf{poly}(\lambda))$. Therefore, the rate of the proposed scheme is

$$\frac{r \cdot |m|}{(|m| + S + \mathsf{poly}(\lambda))} = \frac{r}{\left(1 + \frac{S}{|m|} + \frac{\mathsf{poly}(\lambda)}{|m|}\right)}$$

Size of the secret keys

The size of the secret key associated with a function f is $|\mathsf{fe.sk}_f|$ where $\mathsf{fe.sk}_f$ is an FE secret key for the circuit $C_{f,\mathsf{ske.ct},\mathsf{ske.ct}'}$. To use this secret key, the decryption algorithm would require the entire description of $C_{f,\mathsf{ske.ct},\mathsf{ske.ct}'}$ (see Theorem 9 and Theorem 10).

In addition to the description of f and the decryption circuits of SKE and IncSKE, the circuit $C_{f,\mathsf{ske.ct},\mathsf{ske.ct}'}$ contains two SKE ciphertexts generated during the key generation process. Therefore, $\mathsf{fe.sk}_f$ should contain both an FE secret key for the circuit $C_{f,\mathsf{ske.ct},\mathsf{ske.ct}'}$ and the two ciphertexts $\mathsf{ske.ct}, \mathsf{ske.ct}'$. The first SKE ciphertext is an encryption of an incompressible SKE scheme's secret key $\mathsf{inc.sk}$ along with a bit $flag$, and the second is just a bit y. Using Dziembowski's construction (see Theorem 7), we have $|\mathsf{inc.sk}| = \mathsf{poly}(\lambda)$. Therefore, $|\mathsf{fe.sk}_f|$ is the size of an FE secret key associated with $C_{f,\mathsf{ske.ct},\mathsf{ske.ct}'}$, with an additional $\mathsf{poly}(\lambda)$.

▶ **Theorem 18.** *Assuming* $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.Keygen}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ *is an* ct*-rate-*r *adaptively (selectively) secure FE scheme,* $\mathsf{SKE} = (\mathsf{SKE.Setup}, \mathsf{SKE.Enc}, \mathsf{SKE.Dec})$ *is a secure SKE scheme against unbounded number of ciphertexts and* $\mathsf{IncSKE} = (\mathsf{IncSKE.Setup}, \mathsf{IncSKE.Enc}, \mathsf{IncSKE.Dec})$ *be a secure incompressible SKE scheme, then the above construction is an adaptively (selectively) secure semi-strongly incompressible FE scheme. Also,*

$$|\mathsf{mpk}| = |\mathsf{fe.mpk}|, \quad |\mathsf{sk}| = |\mathsf{fe.sk}|, \quad |\mathsf{ct}| = (|m| + S + \mathsf{poly}(\lambda))/r$$

Proof-Sketch. The proof proceeds via a sequence of hybrid experiments.

- H_0: This corresponds to the adaptive semi-strong incompressibility security game. The adversary first receives the master public key mpk. It queries for polynomially many non-distinguishing keys, and receives the corresponding secret keys. Then, it sends its challenge messages m_0, m_1 and receives the challenge ciphertext which is encryption of m_b. Again, the adversary queries for polynomially many non-distinguishing keys, and receives the corresponding secret keys. It then compresses its state, and queries for both distinguishing and non-distinguishing functions. Upon receiving the secret keys, it must guess which message was encrypted.
- H_1 : In this hybrid, the challenger keeps the challenge ciphertext same as before, but modifies the *distinguishing* secret keys. For every *distinguishing* key query for some function f, it encrypts $f(m_b)$ instead of 0 to generate $\mathsf{ske.ct}'$.
  Note that H_0 and H_1 are indistinguishable due to SKE security. This is because the attacker does not get access to the SKE secret key, except in the form of ciphertexts as part of functional secret keys. We highlight we know the value $f(m_b)$ because the adversary is allowed to make distinguishing key queries in the second phase only.

- H_2 : In this hybrid, the challenger generates $\mathsf{inc.ct}^*$ by encrypting $\mathsf{ske.sk}'$ instead of sampling a truly random string. The rest of the game proceeds as before.
  Indistinguishability of hybrids H_1 and H_2 follows from standard indistinguishability-based security of the incompressible SKE scheme. This is because the secret key $\mathsf{inc.sk}$ is used only during the generation of $\mathsf{inc.ct}^*$.
- H_3 : In this hybrid experiment, the challenger keeps the challenge ciphertext same as before, but modifies the *distinguishing* secret keys. For every *distinguishing* key query for some function f, it encrypts $(1, \mathsf{inc.sk})$ instead of $(0, 0^\alpha)$ to generate $\mathsf{ske.ct}$.
  Note that H_2 and H_3 are indistinguishable due to SKE security. This is because the attacker does not get access to the SKE secret key, except in the form of ciphertexts as part of functional secret keys.
- H_4 : In this hybrid, the challenger modifies the challenge ciphertext ct^* by encrypting $(m_0, \mathsf{ske.sk}, \mathsf{inc.ct}^*, 1)$ instead of $(m_b, \bot, \mathsf{inc.ct}^*, 0)$. The rest of the game proceeds as before.
  Indistinguishability of hybrids H_3 and H_4 follows from adaptive indistinguishability-based security of FE, as the output of the function stays the same for all key queries. This is because in H_4, the non-distinguishing keys would decrypt the challenge ciphertext to $f(m_0)$ (since b is set to 1 and $flag = 0$), which is equal to $f(m_b)$. Whereas, the distinguishing keys would decrypt to $f(m_b)$ (since b is set to 1 and $flag = 1$, see Figure 1).
- H_5 : In this hybrid, the challenger reverts to generating $\mathsf{inc.ct}^*$ by sampling a truly random string. The rest of the game proceeds as before.
  Indistinguishability of hybrids H_4 and H_5 follows from the *incompressible* indistinguishability-based security of the incompressible SKE scheme. This is because the information $\mathsf{inc.sk}$ is needed to generate distinguishing keys in the second phase.
  Finally, note that in H_5, the bit b is used only in the second phase while generating distinguishing keys. That is, $\mathsf{ske.ct}'$ is generated by encrypting $f(m_b)$. Therefore, the winning probability for any adversary in G_5 depends on the standard indistinguishability-based security of the regular SKE scheme. ◀

The proof for the above theorem is provided in the full version [18]. Using Theorem 9, we have the following corollary.

▶ **Corollary 19.** *Assuming the existence of selectively secure FE for circuits, there exists a* ct*-rate-*$\frac{1}{4}$ *adaptively secure semi-strongly incompressible FE scheme for circuits. Also,*

$$|\mathsf{mpk}| = \mathsf{poly}(\lambda), \quad |\mathsf{sk}_f| = \mathsf{poly}(\lambda), \quad |\mathsf{ct}| = 2(|m| + S) + \mathsf{poly}(\lambda)$$

Using Theorem 10, we have the following corollary.

▶ **Corollary 20.** *Assuming the existence of selectively secure FE for circuits, there exists a* ct*-rate-*$\frac{1}{2}$ *selectively secure semi-strongly incompressible FE scheme for circuits. Also,*

$$|\mathsf{mpk}| = \mathsf{poly}(\lambda), \quad |\mathsf{sk}_f| = \mathsf{poly}(\lambda), \quad |\mathsf{ct}| = |m| + S + \mathsf{poly}(\lambda)$$

5 Rate-1 Incompressible FE with Large Keys and Semi-Strong Security

In this section, we present a *selective semi-strong incompressible* FE scheme, using a (regular) public key FE, together with other standard cryptographic primitives. If the underlying FE scheme has ct-rate-1, then our incompressible FE scheme also has ct-rate-1. Although the functional key sizes are larger.

Our construction

For simplicity, we consider boolean-valued functions. To design an incompressible FE supporting function class $\{\mathcal{F}_n : \{0,1\}^n \to \{0,1\}\}_{n\in\mathbb{N}}$, we use the following primitives. Throughout S denotes the (in)compressibility parameter.

- $\mathsf{Ext} : \{0,1\}^d \times \{0,1\}^\ell \to \{0,1\}^\lambda$ be a strong average-min entropy randomness extractor, where $d = d(\lambda)$ and $\ell = \ell(\lambda, S)$.
- $\mathsf{SKE} = (\mathsf{SKE.Setup}, \mathsf{SKE.Enc}, \mathsf{SKE.Dec})$ be an SKE scheme that can encrypt messages of length $n + d + 1$. (Here n is the FE message length and d is the seed length. W.l.o.g., we assume the secret key length to be λ.)
- $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.Keygen}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ be an FE scheme with message spaces $\{\{0,1\}^n\}_n$ and function space $\left\{\tilde{\mathcal{F}}_n\right\}_n$.
  Let $\tilde{n} = \max(n, S) + \lambda + 1$. We require the function class $\tilde{\mathcal{F}}_{\tilde{n}}$ to be such that it contains $\mathcal{F}_n$, and the decryption circuit of SKE and extractor Ext.

We describe the algorithms for our incompressible FE scheme below.

- $\mathsf{Setup}(1^\lambda, 1^S, 1^n)$: Let $\tilde{n} = \max(n, S) + \lambda + 1$. The setup algorithm samples a master public and secret key of the FE scheme by sampling $(\mathsf{fe.msk}, \mathsf{fe.mpk}) \leftarrow \mathsf{FE.Setup}(1^\lambda, 1^{\tilde{n}})$. It also samples an SKE secret key $\mathsf{ske.sk} \leftarrow \mathsf{SKE.Setup}(1^\lambda)$. It sets $\mathsf{msk} = (\mathsf{ske.sk}, \mathsf{fe.msk})$ and $\mathsf{mpk} = \mathsf{fe.mpk}$.
- $\mathsf{KeyGen}(\mathsf{msk}, f)$: Parse master key as $\mathsf{msk} = (\mathsf{ske.sk}, \mathsf{fe.msk})$. The key generation algorithm first computes ciphertext $\mathsf{ske.ct} = \mathsf{SKE.Enc}(\mathsf{ske.sk}, \mathbf{0})$. That is, it encrypts $n+d+1$ zeros. Then, it generates an FE secret key by computing $\mathsf{fe.sk}_f \leftarrow \mathsf{FE.Keygen}(\mathsf{fe.msk}, C_{f,\mathsf{ske.ct}})$, where C is described in Figure 2.

Input: $x \in \{0,1\}^{\max(n,S)}$, an SKE secret key $\mathsf{ske.sk} \in \{0,1\}^\lambda$, bit $\mathsf{flag} \in \{0,1\}$.
Hardwired: Function f, an SKE ciphertext $\mathsf{ske.ct}$.
Output: $y \in \{0,1\}$.

1. Check if $\mathsf{flag} = 0$:
   – If yes, parse x as an n-bit message m, and output $f(m)$.
   That is, if $S > n$, then $m = x[1:n]$, else $m = x$.
2. Otherwise, compute $(b, s, v) = \mathsf{SKE.Dec}(\mathsf{ske.sk}, \mathsf{ske.ct})$.
   Here $b \in \{0,1\}$, $s \in \{0,1\}^d, v \in \{0,1\}^n$.
3. Check if $b = 0$:
   – If yes, then output first bit of s, i.e. $s[1]$.
4. Otherwise, parse x as an S-bit source R, and output $f(v \oplus \mathsf{Ext}_s(R))$.
   That is, if $S < n$, then set $R = x[1:S]$, else set $R = x$.
   Then compute message $m = v \oplus \mathsf{Ext}_s(R)$ and output $f(m)$.

Figure 2 Description of $C_{f,\mathsf{ske.ct}}$.

- $\mathsf{Enc}(\mathsf{msk}, m)$: The encryption algorithm generates the ciphertext by computing $\mathsf{fe.ct} \leftarrow \mathsf{FE.Enc}(\mathsf{msk}, (m, \perp, 0))$. It outputs $\mathsf{fe.ct}$. (We point out that the message m is padded to be of length S in case $S > n$. Also, by $\perp$, we mean it sets a null secret key.)
- $\mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct})$: The decryption algorithm decrypts the ciphertext by computing $m = \mathsf{FE.Dec}(\mathsf{sk}_f, \mathsf{ct})$. It returns m.

Correctness

Follows immediately from the construction.

Size of the master public key

The master public key of our scheme is $\mathsf{fe.mpk}$. Therefore, the size is $|\mathsf{fe.mpk}|$.

Size of the ciphertext

The size of the message encrypted by $\mathsf{FE.Enc}$ is $\tilde{n} = \max(n, S) + \lambda + 1$. Now if the base FE scheme has ct-rate-1, then ciphertext size in our scheme is also $\max(S, n) + \mathsf{poly}(\lambda)$.

Size of the secret key

The size of the secret key associated with a function f is $|\mathsf{fe.sk}_f|$ where $\mathsf{fe.sk}_f$ is an FE secret key for the circuit $C_{f,\mathsf{ske.ct}}$. To use this secret key, the decryption algorithm would require the entire description of $C_{f,\mathsf{ske.ct}}$ (see Theorem 10).

In addition to the description of f and the decryption circuit of SKE, the circuit $C_{f,\mathsf{ske.ct}}$ contains an SKE ciphertext generated during the key generation process. So, $\mathsf{fe.sk}_f$ should contain both an FE secret key for the circuit $C_{f,\mathsf{ske.ct},\mathsf{ske.ct}'}$ and the ciphertext $\mathsf{ske.ct}$. The SKE ciphertext is an encryption of a message of length $n + d + 1$ where d is the seed length of the extractor[11] and n is the length of the message. Therefore, $|\mathsf{fe.sk}_f|$ is equal to $n + d + 1$ plus the size of an FE secret key associated with $C_{f,\mathsf{ske.ct}}$.

▶ **Theorem 21.** *Assuming* $\mathsf{SKE} = (\mathsf{SKE.Setup}, \mathsf{SKE.Enc}, \mathsf{SKE.Dec})$ *is IND-CPA secure,* $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.Keygen}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ *is a selectively secure FE with* ct*-rate-*r*, then the above FE construction is a selective semi-strong incompressible FE scheme. Also,*

$$|\mathsf{mpk}| = |\mathsf{fe.mpk}|, \quad |\mathsf{sk}| = |\mathsf{fe.sk}| + |\mathsf{ske.ct}|, \quad |\mathsf{ct}| = (\max(n, S) + \mathsf{poly}(\lambda))/r$$

Proof-Sketch. The proof proceeds via a sequence of hybrid experiments.

- H_0: This corresponds to the selective semi-strong incompressibility security game. The adversary first sends its challenge messages m_0, m_1 and receives the challenge ciphertext which is encryption of m_b. Next, the adversary queries for polynomially many non-distinguishing keys, and receives the corresponding secret keys. It then compresses its state, and queries for both distinguishing and non-distinguishing functions. Upon receiving the secret keys, it must guess which message was encrypted.
- H_1 : In this hybrid, the challenger keeps the challenge ciphertext same as before, but modifies the secret keys. During setup, the challenger samples a *single* random string $R \leftarrow \{0,1\}^S$ as the extractor source, and a *single* random string $s \leftarrow \{0,1\}^d$ as the extractor seed. Now, instead of using an SKE encryption of $\mathbf{0}$ within each secret key, the challenger does the following:
  - For every *non-distinguishing* key query for some function f, it encrypts $(0, f(m_0), \mathbf{0})$. That is, the second message bit is $f(m_0)$ and rest are still all zeros.
  - For every *distinguishing* key query for some function f, it sets $v = m_b \oplus \mathsf{Ext}_s(R)$, and computes the SKE ciphertext as to be an encryption of $(1, s, v)$.

[11] d is equal to $\mathsf{poly}(\lambda)$, see Theorem 2

Note that H_0 and H_1 are indistinguishable due to SKE security. This is because the attacker does not get access to the SKE secret key, except in the form of ciphertexts as part of functional secret keys. We highlight that since we need to know $f(m_0)$ to answer even a non-distinguishing key query, thus we can only prove selective security of our incompressible FE scheme.

- H_2 : In this hybrid, the challenger encrypts $(R, \mathsf{ske.sk}, 1)$ instead of $(m_b, \perp, 0)$. (Note that here we are overloading notation and consider that there is a fixed deterministic way, e.g. by padding, to write R and m as $\max(n, S)$-bit strings.) The rest of the game proceeds as before.
  Indistinguishability of hybrids H_1 and H_2 follows standard selective indistinguishability-based security of FE, as the output of the function stays the same for all key queries. This is because for non-distinguishing keys we know that $f(m_0) = f(m_b)$, while for non-distinguishing keys, the function output is still $f(m_b)$.
- H_3 : In this hybrid experiment, v is chosen uniformly at random (instead of setting it as $m_b \oplus \mathsf{Ext}_s(R)$. Using the strong extractability guarantee of the extractor, we can argue that H_2 and H_3 are negligibly close.
  Finally, note that the bit b is not used in H_3, and therefore the adversary has no advantage in this experiment. ◀

The detailed proof for the above theorem is provided in the full version [18]. Using Theorem 10, we have the following corollary.

▶ **Corollary 22.** *Assuming the existence of* ct*-rate-*$\frac{1}{2}$ *selectively secure FE for circuits, there exists a* ct*-rate-*$\frac{1}{2}$ *selectively secure semi-strongly incompressible FE scheme for circuits. Also,*

$$|\mathsf{mpk}| = \mathsf{poly}(\lambda), \quad |\mathsf{sk}_f| = |m| + \mathsf{poly}(\lambda), \quad |\mathsf{ct}| = |m| + S + \mathsf{poly}(\lambda)$$

6 Rate-1 Incompressible FE with Short Keys and Standard Security

In this section, we present a *selective (standard) incompressible* FE scheme, using a (regular) public key FE, together with other standard cryptographic primitives. If the underlying FE scheme has ct-rate-1, then our incompressible FE scheme also has ct-rate-1. Interestingly, we show that if the underlying FE scheme has short keys, then our resulting incompressible FE scheme has short keys as well. While it might seem to contradict the lower bound [3], we observe that since we only prove standard incompressibility security (where only one non-distinguishing key gets corrupted) and the functions supported by our FE scheme has single-bit output, thus this does not contradict the result. This is because the lower bound actually states that if an incompressible FE scheme has ct-rate-1, then the size of the functional secret key must grow with the output length of the function, and not necessarily the input/message length.

Our construction

As discussed in the overview, this construction relies on the same ingredients as our FE construction from Section 5. The core difference is that now the functions that we create secret keys for change slightly. Below we describe the algorithms where we make modifications, and underline all the changes.

- $\mathsf{Setup}(1^\lambda, 1^S, 1^n)$: This is the same as in Section 5.

- $\mathsf{KeyGen}(\mathsf{msk}, f)$: Parse master key as $\mathsf{msk} = (\mathsf{ske.sk}, \mathsf{fe.msk})$. The key generation algorithm first computes ciphertext $\mathsf{ske.ct} = \mathsf{SKE.Enc}(\mathsf{ske.sk}, \underline{\mathbf{0}})$. That is, it encrypts $d+2$ zeros. Then, it generates an FE secret key by computing $\mathsf{fe.sk}_f \leftarrow \mathsf{FE.Keygen}(\mathsf{fe.msk}, C'_{f,\mathsf{ske.ct}})$, where C is described in Figure 3.

Input: $x \in \{0,1\}^{\max(n,S)}$, an SKE secret key $\mathsf{ske.sk} \in \{0,1\}^{\lambda}$, bit $\mathsf{flag} \in \{0,1\}$.
Hardwired: Function f, an SKE ciphertext $\mathsf{ske.ct}$.
Output: $y \in \{0,1\}$.

1. Check if $\mathsf{flag} = 0$:
   – If yes, parse x as an n-bit message m, and output $f(m)$.
2. Otherwise, compute $(b, s, v) = \mathsf{SKE.Dec}(\mathsf{ske.sk}, \mathsf{ske.ct})$.
   Here $b \in \{0,1\}, s \in \{0,1\}^d, \underline{v \in \{0,1\}}$.
3. Check if $b = 0$:
   – If yes, then output first bit of s, i.e. $s[1]$.
4. Otherwise, parse x as an S-bit source R, and output $\underline{v \oplus \mathsf{Ext}_s(R)}$.
   That is, it no longer computes f in this case.

Figure 3 Description of $C'_{f,\mathsf{ske.ct}}$.

- $\mathsf{Enc}(\mathsf{msk}, m), \mathsf{Dec}(\mathsf{sk}_f, \mathsf{ct})$: The encryption and decryption algorithms are the same as in Section 5.

Correctness

Follows immediately from the construction.

Size of the master public key and ciphertext

This is the same as in Section 5.

Size of the secret key

The size of the secret key associated with a function f is $|\mathsf{fe.sk}_f|$ where $\mathsf{fe.sk}_f$ is an FE secret key for the circuit $C'_{f,\mathsf{ske.ct}}$. To use this secret key, the decryption algorithm would require the entire description of $C'_{f,\mathsf{ske.ct}}$ (see Theorem 10).

In addition to the description of f and the decryption circuit of SKE, the circuit $C_{f,\mathsf{ske.ct}}$ contains an SKE ciphertext generated during the key generation process. So, $\mathsf{fe.sk}_f$ should contain both an FE secret key for the circuit $C_{f,\mathsf{ske.ct},\mathsf{ske.ct}'}$ and the ciphertext $\mathsf{ske.ct}$. The SKE ciphertext is an encryption of a message of length $d+2$ where d is the seed length of the extractor[12]. Therefore, $|\mathsf{fe.sk}_f|$ is the size of an FE secret key associated with $C_{f,\mathsf{ske.ct}}$, with an addition $d+1$.

▶ **Theorem 23.** *Assuming* $\mathsf{SKE} = (\mathsf{SKE.Setup}, \mathsf{SKE.Enc}, \mathsf{SKE.Dec})$ *is IND-CPA secure,* $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.Keygen}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ *is a selectively secure FE with* ct*-rate-*r*, then the above FE construction is a selective (standard) incompressible FE scheme. Also,*

$$|\mathsf{mpk}| = |\mathsf{fe.mpk}|, \quad |\mathsf{sk}| = |\mathsf{fe.sk}|, \quad |\mathsf{ct}| = (\max(n, S) + \mathsf{poly}(\lambda))/r$$

[12] d is equal to $\mathsf{poly}(\lambda)$, see Theorem 2

Proof-Sketch. The proof is very similar to the security proof of Theorem 21, except with one change. Since we only prove standard incompressibility of our FE scheme, then we only need to answer a single distinguishing key query. Therefore, rather than programming v as $m_b \oplus \mathsf{Ext}_s(R)$ as in the previous proof, we simply program $v = f(m_b) \oplus \mathsf{Ext}_s(R)$. That is, we only hardwire one function value inside v rather than the full message m_b. Because of this change, we can only prove standard incompressibility security (because we cannot answer more than one non-distinguishing key), but the circuit $C'_{f,\mathsf{ske.ct}}$ (that we have to create a functional key for) now has a succinct description as it only needs f and $\mathsf{ske.ct}$ (which is short anyways). For completeness, we sketch the hybrids below.

- H_0: This corresponds to the selective standard incompressibility security game. The adversary first sends its challenge messages m_0, m_1 and receives the challenge ciphertext which is encryption of m_b. Next, the adversary queries for polynomially many non-distinguishing keys, and receives the corresponding secret keys. It then compresses its state, and queries for one distinguishing function $f\mathsf{st}$ and many non-distinguishing functions. Upon receiving the secret keys, it must guess which message was encrypted.
- H_1 : In this hybrid, the challenger keeps the challenge ciphertext same as before, but modifies the secret keys. During setup, the challenger samples a single random string $R \leftarrow \{0,1\}^S$ as the extractor source, and a random string $s \leftarrow \{0,1\}^d$ as the extractor seed. Now, instead of using an SKE encryption of $\mathbf{0}$ within each secret key, the challenger does the following:
  - For every *non-distinguishing* key query for some function f, it encrypts $(0, f(m_0), \mathbf{0})$. That is, the second message bit is $f(m_0)$ and rest are still all zeros.
  - For *distinguishing* key query f, it sets $v = f(m_b) \oplus \mathsf{Ext}_s(R)$, and computes the SKE ciphertext as to be an encryption of $(1, s, v)$.
- H_2 : In this hybrid, the challenger encrypts $(R, \mathsf{ske.sk}, 1)$ instead of $(m_b, \perp, 0)$. The rest of the game proceeds as before.
- H_3 : In this hybrid experiment, v is chosen uniformly at random (instead of setting it as $f(m_b) \oplus \mathsf{Ext}_s(R)$.

  Finally, note that the bit b is not used in H_3, and therefore the adversary has no advantage in this experiment. ◀

The detailed proof for the above theorem is provided in the full version [18]. Using Theorem 10, we have the following corollary.

▶ **Corollary 24.** *Assuming the existence of* ct*-rate-*1 *selectively secure FE for circuits, there exists a* ct*-rate-*1 *selectively secure regular incompressible FE scheme for circuits. Also,*

$$|\mathsf{mpk}| = \mathsf{poly}(\lambda), \quad |\mathsf{sk}_f| = \mathsf{poly}(\lambda), \quad |\mathsf{ct}| = \max(S, n) + \mathsf{poly}(\lambda)$$

▶ Remark 25 (Handling functions with longer outputs, or a fixed number of distinguishing keys). The above FE construction can be naturally extended to support functions with longer outputs, or a fixed number of distinguishing keys. The idea is quite simply to hardwire either the multi-bit function output (in the first case) or the function output for every distinguishing function (in the second case). Note that if the secret keys for the underlying FE scheme are short (independent of the function description), then the above scheme achieves incompressible FE with optimal ciphertexts as well as secret keys.

References

1 Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. In *CCS '12*, 2012. doi:10.1145/2382196.2382279.

2 John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334, 2007. doi:10.1109/SP.2007.11.

3 Kaartik Bhushan, Rishab Goyal, Venkata Koppula, Varun Narayanan, Manoj Prabhakaran, and Mahesh Sreekumar Rajasree. Leakage-resilient incompressible cryptography: Constructions and barriers. In *Advances in Cryptology–ASIACRYPT*, 2024.

4 Nir Bitansky and Tomer Solomon. Bootstrapping homomorphic encryption via functional encryption. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 17:1–17:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ITCS.2023.17.

5 Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, 2001.

6 Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit abe and compact garbled circuits. In *Advances in Cryptology–EUROCRYPT 2014: 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings 33*, pages 533–556. Springer, 2014. doi:10.1007/978-3-642-55220-5_30.

7 Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: definitions and challenges. In *TCC*, 2011.

8 Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In *TCC*, 2007.

9 Pedro Branco, Nico Döttling, and Jesko Dujmović. Rate-1 Incompressible Encryption from Standard Assumptions. In Eike Kiltz and Vinod Vaikuntanathan, editors, *Theory of Cryptography*, Lecture Notes in Computer Science, pages 33–69, Cham, 2022. Springer Nature Switzerland. doi:10.1007/978-3-031-22365-5_2.

10 Chongwon Cho, Nico Döttling, Sanjam Garg, Divya Gupta, Peihan Miao, and Antigoni Polychroniadou. Laconic oblivious transfer and its applications. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017 – 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 33–65. Springer, 2017. doi:10.1007/978-3-319-63715-0_2.

11 Clifford Cocks. An identity based encryption scheme based on Quadratic Residues. In *Cryptography and Coding, IMA International Conference*, volume 2260 of LNCS, pages 360–363, 2001. doi:10.1007/3-540-45325-3_32.

12 Whitfield Diffie and Martin E. Hellman. New directions in cryptography, 1976.

13 Nico Döttling, Phillip Gajland, and Giulio Malavolta. Laconic function evaluation for turing machines. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *Public-Key Cryptography – PKC 2023 – 26th IACR International Conference on Practice and Theory of Public-Key Cryptography, Atlanta, GA, USA, May 7-10, 2023, Proceedings, Part II*, volume 13941 of *Lecture Notes in Computer Science*, pages 606–634. Springer, 2023. doi:10.1007/978-3-031-31371-4_21.

14 Stefan Dziembowski. On Forward-Secure Storage. In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, Lecture Notes in Computer Science, pages 251–270, Berlin, Heidelberg, 2006. Springer. doi:10.1007/11818175_15.

15 Sanjam Garg, Omkant Pandey, and Akshayaram Srinivasan. Revisiting the cryptographic hardness of finding a nash equilibrium. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016 – 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 579–604. Springer, 2016. doi:10.1007/978-3-662-53008-5_20.

16 Shafi Goldwasser, Yael Kalai, Raluca Ada Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 555–564, 2013.

17 Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology – CRYPTO 2015 – 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 503–523. Springer, 2015. doi:10.1007/978-3-662-48000-7_25.

18 Rishab Goyal, Venkata Koppula, Mahesh Sreekumar Rajasree, and Aman Verma. Incompressible functional encryption. Cryptology ePrint Archive, Paper 2024/798, 2024. URL: https://eprint.iacr.org/2024/798.

19 Rishab Goyal, Venkata Koppula, Andrew Russell, and Brent Waters. Risky traitor tracing and new differential privacy negative results. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018 – 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 467–497. Springer, 2018. doi:10.1007/978-3-319-96884-1_16.

20 Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017*, pages 612–621, 2017. doi:10.1109/FOCS.2017.62.

21 Rishab Goyal, Venkata Koppula, and Brent Waters. Collusion resistant traitor tracing from learning with errors. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 660–670. ACM, 2018. doi:10.1145/3188745.3188844.

22 Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 – November 3, 2006*, pages 89–98. ACM, 2006. doi:10.1145/1180405.1180418.

23 Jiaxin Guan, Daniel Wichs, and Mark Zhandry. Incompressible Cryptography. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022*, Lecture Notes in Computer Science, pages 700–730, Cham, 2022. Springer International Publishing. doi:10.1007/978-3-031-06944-4_24.

24 Jiaxin Guan, Daniel Wichs, and Mark Zhandry. Multi-instance randomness extraction and security against bounded-storage mass surveillance. In *Theory of Cryptography Conference*, pages 93–122. Springer, 2023. doi:10.1007/978-3-031-48621-0_4.

25 Aayush Jain, Huijia Lin, and Ji Luo. On the optimal succinctness and efficiency of functional encryption and attribute-based encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 479–510. Springer, 2023. doi:10.1007/978-3-031-30620-4_16.

26 Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In *STOC*, 2021.

27 Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from lpn over f p, dlin, and prgs in nc 0. In *Advances in Cryptology–EUROCRYPT 2022: 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30–June 3, 2022, Proceedings, Part I*, pages 670–699. Springer, 2022.

28 Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT*, 2008.

29 Lucas Kowalczyk, Tal Malkin, Jonathan Ullman, and Daniel Wichs. Hardness of non-interactive differential privacy from one-way functions, 2018.

30 Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010.

31 Willy Quach, Hoeteck Wee, and Daniel Wichs. Laconic function evaluation and applications. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 859–870. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00086.

32 Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005. doi:10.1007/11426639_27.

33 Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, 1984.

34 Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017*, pages 600–611, 2017. doi:10.1109/FOCS.2017.61.

35 Andrew Yao. How to generate and exchange secrets. In *FOCS*, 1986.

Deterministic Approximation for the Volume of the Truncated Fractional Matching Polytope

Heng Guo
School of Informatics, University of Edinburgh, UK

Vishvajeet N
School of Informatics, University of Edinburgh, UK

Abstract

We give a deterministic polynomial-time approximation scheme (FPTAS) for the volume of the truncated fractional matching polytope for graphs of maximum degree Δ, where the truncation is by restricting each variable to the interval $[0, \frac{1+\delta}{\Delta}]$, and $\delta \leq \frac{C}{\Delta}$ for some constant $C > 0$. We also generalise our result to the fractional matching polytope for hypergraphs of maximum degree Δ and maximum hyperedge size k, truncated by $[0, \frac{1+\delta}{\Delta}]$ as well, where $\delta \leq C\Delta^{-\frac{2k-3}{k-1}} k^{-1}$ for some constant $C > 0$. The latter result generalises both the first result for graphs (when $k = 2$), and a result by Bencs and Regts (2024) for the truncated independence polytope (when $\Delta = 2$). Our approach is based on the cluster expansion technique.

2012 ACM Subject Classification Theory of computation → Random walks and Markov chains; Theory of computation → Computational geometry; Theory of computation → Pseudorandomness and derandomization

Keywords and phrases deterministic volume computation, cluster expansion, explicit polytopes

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.57

Related Version *Full Version*: `https://arxiv.org/abs/2409.07283`

Funding This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 947778).

1 Introduction

The evaluation of volume is a classic computing task. The pioneer work of Dyer, Frieze, and Kannan [11] showed that given a membership oracle, the volume of a convex body can be approximated to arbitrary accuracy in randomised polynomial time. On the other hand, in the same model, deterministic approximation takes at least exponential time [12, 2]. This marked an early milestone that distinguished the computational power of deterministic algorithms from that of randomized algorithms.

Nonetheless, computing volumes might still be interesting for other models and/or more restricted cases, where the lower bound of [12, 2] does not apply, and interesting deterministic volume algorithms may still exist. In particular, there appears to be no run-time lower bound for deterministically approximating the volume of a polytope given by facets. In this case, as the membership oracle is trivial to implement, the randomised algorithm of Dyer, Frieze, and Kannan [11] achieves an ε-approximation in time polynomial in the input size and $\frac{1}{\varepsilon}$. The deterministic counterpart of such an algorithm is called *fully polynomial-time approximation scheme* (FPTAS). Unfortunately, the best deterministic algorithmic technique for polytope volumes appears to fall short of achieving FPTAS. They either require exponential time [18, 5][1] or yield exponential approximation [3, 4]. In the hope of getting an FPTAS, we may focus on even more restricted cases, such as the independence polytope or matching polytope for graphs.

[1] The algorithms of [18, 5] are in fact exact algorithms and the dominating term in their run-time is linear in the number of vertices of the polytope.


© Heng Guo and Vishvajeet N;
licensed under Creative Commons License CC-BY 4.0

Along this direction, Gamarnik and Smedira [13] considered the relaxed independence polytope. Given a graph $G = (V, E)$, it is defined as $\{\mathbf{x} \in [0,1]^V \mid \forall \{u, v\} \in E,\ x_u + x_v \leq 1\}$. Clearly, if we further restrict every x_v to the interval $[0, 1/2]$, this polytope degenerates to a cube of side length $1/2$ and its volume is simply $2^{-|V|}$. Gamarnik and Smedira [13] showed that we can push beyond this trivial case, namely, if we truncate the relaxed independence polytope by the interval $\left[0, \frac{1}{2}\left(1 + \frac{O(1)}{\Delta^2}\right)\right]$ for graphs of maximum degree Δ, a quasi-polynomial-time approximation scheme exists. Their technique is based on the correlation decay approach [22, 1]. Subsequently, Bencs and Regts [8] improved the interval to $\left[0, \frac{1}{2}\left(1 + \frac{O(1)}{\Delta}\right)\right]$ and the run-time to polynomial-time, based on the zeros of polynomial approach [6, 20]. The bound on this interval appears to be where the limit of the current methods is.

In this paper, we explore what other polytopes may admit efficient deterministic volume approximation. In particular, we consider the natural dual of the independence polytope, namely the matching polytope, or more precisely, its standard relaxation, the fractional matching polytope. Note that although it is the dual of the relaxed independence polytope, its volume might be drastically different.

▶ **Definition 1** (Fractional matching polytope). *Given a graph $G = (V, E)$, the* fractional matching polytope *is defined as follows*

$$P_G := \left\{ \mathbf{x} \in [0,1]^E \mid \sum_{e \sim v} x_e \leq 1 \text{ for every } v \in V \right\},$$

where $e \sim v$ if the edge e is adjacent to v.

For the fractional matching polytope, the trivial truncation is with the interval $\left[0, \frac{1}{\Delta}\right]$. Similar to [13, 8], we also truncate P_G by an interval that is slightly longer, multiplicatively, than the trivial truncation.

▶ **Definition 2** (Truncated fractional matching polytope). *For $\delta > 0$ and a graph $G = (V, E)$ of maximum degree Δ, the* truncated fractional matching polytope *is defined as follows*

$$P_{G,\delta} := \left\{ \mathbf{x} \in \left[0, \frac{1+\delta}{\Delta}\right]^E \mid \sum_{e \sim v} x_e \leq 1 \text{ for every } v \in V \right\}.$$

Denote the interval $\left[0, \frac{1+\delta}{\Delta}\right]$ by M_δ. Then $P_{G,\delta} = {M_\delta}^E \cap P_G$. Additionally, for any $v \in V$, the constraint $\sum_{e \sim v} x_e \leq 1$ is denoted by C_v. We are interested in computing the volume of $P_{G,\delta}$, i.e. the quantity

$$\mathrm{Vol}(P_{G,\delta}) := \int_{{M_\delta}^E} \prod_{v \in V} \mathbb{1}_{C_v} d\mu, \tag{1}$$

where μ is the Lebesgue measure and the indicator function $\mathbb{1}_{C_v} : {M_\delta}^E \to \{0, 1\}$ outputs 1 if and only if the constraint C_v is satisfied, i.e. when $\sum_{e \sim v} x_e \leq 1$.

Our main result regarding $\mathrm{Vol}(P_{G,\delta})$ is the following. Interestingly, similar to [8], the relative margin for the truncation interval we get is also $\frac{C}{\Delta}$.

▶ **Theorem 3.** *For graphs of maximum degree $\Delta \geq 2$ and $\delta \leq \frac{C}{\Delta}$ for some constant $C > 0$, there is a fully polynomial-time approximation scheme (FPTAS) for* $\mathrm{Vol}(P_{G,\delta})$.

The proof of Theorem 3 relies on the algorithmic cluster expansion approach [14, 16]. This is very closely related to the zeros of polynomial approach Bencs and Regts [8] took. The first step is to rewrite $\mathrm{Vol}(P_{G,\delta})$ as the partition function of a polymer model on the graph G. Roughly speaking, each polymer corresponds to a set of connected vertices where the constraints on these vertices are violated. For the algorithmic approach of the polymer model to work, we need to show that the total weights of these violations decay rapidly, or more technically, the so-called Kotecký-Preiss criterion [17] holds. Intuitively, when we truncate with the trivial interval $\left[0, \frac{1}{\Delta}\right]$, none of the violation may happen. With a longer truncation interval, violations can happen, but the small relative margin we allow implies that violations can only happen with small probability. Our method involves some careful estimates of these violation probabilities. We also note that it appears difficult to apply the correlation decay method here, and a direct application of the argument of Bencs and Regts [8] would yield a smaller margin $\delta = \frac{C}{\Delta^2}$ for some constant $C > 0$.

As mentioned earlier, the volumes of the relaxed independence polytope and the fractional matching polytope do not appear to be related, and thus Theorem 3 is not directly comparable to the main result of [8]. To put both results under a unified framework, we also consider hypergraph matchings. Let $H = (V, E)$ be a hypergraph. Consider the following polytope:

$$P_H := \left\{ \mathbf{x} \in [0,1]^E \mid \sum_{e \ni v} x_e \leq 1 \text{ for every } v \in V \right\}, \tag{2}$$

where $e \ni v$ if the hyperedge e contains v. This is a relaxation of the hypergraph matching polytope and a natural generalisation of the fractional matching polytope for graphs in Definition 1. We denote it by the *fractional hypergraph matching polytope*.

We consider hypergraphs of maximum degree $\Delta \geq 2$ and maximum hyperedge size $k \geq 2$. In this case, the polytope P_H is indeed defined by a set of linear constraints where each variable appears at most k times, and each constraint has at most Δ variables. When $k = 2$, this degenerates to the fractional matching polytope, and when $\Delta = 2$, this degenerates to the relaxed independence polytope. Similar to Definition 2, we truncate it by a cube of side length $\frac{1+\delta}{\Delta}$. Recall that M_δ denotes the interval $\left[0, \frac{1+\delta}{\Delta}\right]$. Define the truncated polytope $P_{H,\delta} := {M_\delta}^E \cap P_H$. Our result regarding $\mathrm{Vol}(P_{H,\delta})$ is the following.

▶ **Theorem 4.** *For hypergraphs of maximum degree $\Delta \geq 2$ and maximum hyperedge size $k \geq 2$, and $\delta \leq \frac{C}{\Delta^{\frac{2k-3}{k-1}} k}$ for some constant $C > 0$, there is an FPTAS for $\mathrm{Vol}(P_{H,\delta})$.*

The bound $\delta = O\left(\left(\Delta^{\frac{2k-3}{k-1}} k\right)^{-1}\right)$ recovers Theorem 3. Namely, when $k = 2$, $\delta = O\left(\frac{1}{\Delta}\right)$. Moreover, when $\Delta = 2$, $\delta = O\left(\frac{1}{k}\right)$ and the interval length is $\frac{1+\delta}{\Delta} = \frac{1}{2} + O\left(\frac{1}{k}\right)$, which recovers the result of [8].

The proof of Theorem 4 also relies on the algorithmic cluster expansion approach. It combines our technique for proving Theorem 3 and a generalisation of the technique in [8]. In particular, we introduce a different polymer model in this case, which requires a new generalisation of the classic broken circuit theory [23, 21] to hypergraphs.

An immediate open problem is if Theorem 4 holds with $\delta = O\left(\frac{1}{\Delta k}\right)$, which is discussed in more detail in Section 4. A much more challenging question is if we can obtain deterministic volume approximation for these polytopes without truncation, or with truncation by the interval $[0, 1 - \delta]$ for some small δ. For the closely related problems of approximating the number of independent sets or matchings, deterministic algorithms [22, 20, 7] match their randomised counterparts, at least for bounded degree graphs. In particular, in bounded degree graphs, an efficient deterministic algorithm exists to approximately count the number

of matchings [7]. However, these methods do not seem to carry over to volume approximation. When there is no or little truncation, volume approximation appears to be out of reach for current deterministic methods.

The rest of the paper is organised as follows. In Section 2, we consider the fractional matching polytope and show Theorem 3. In Section 3, we turn our attention to the hypergraph case and show Theorem 4. Finally, in Section 4, we discuss some potential future directions and the current obstacles.

2 Cluster Expansion

In this section, we briefly review the polymer model and the algorithmic cluster expansion approach, and apply it to $\mathrm{Vol}(P_{G,\delta})$ to show Theorem 3. Let $\mathcal{P}$ be a finite set, whose elements we call *polymers*. We endow $\mathcal{P}$ with a symmetric and reflexive incompatibility relation $\not\sim$ between any two polymers, and also a weight function $w(\cdot)$[2] that assigns a weight $w(\gamma)$ to the polymer γ. Given two polymers γ_1 and γ_2, we write $\gamma_1 \not\sim \gamma_2$ if γ_1 and γ_2 are incompatible, and $\gamma_1 \sim \gamma_2$ otherwise. For a set Γ of polymers, we say it is *compatible* if for any two $\gamma_1, \gamma_2 \in \Gamma$, $\gamma_1 \sim \gamma_2$. Additionally, there is a size function for polymers, and we use $|\gamma|$ to denote the size of γ.

▶ **Definition 5** (Polymer partition function). *The partition function of the polymer model above is*

$$\Xi(\mathcal{P}, w) := \sum_{\Gamma} \prod_{\gamma \in \Gamma} w(\gamma),$$

where the sum is over compatible $\Gamma \subseteq \mathcal{P}$.

We will be interested in the cluster expansion which is an infinite series representation of $\log \Xi(P)$. Given a multiset Γ of polymers from $\mathcal{P}$, we define the incompatibility graph $H(\Gamma)$ where we have a vertex v_γ for every polymer $\gamma \in \Gamma$ and an edge between every pair of vertices corresponding to an incompatible pair of polymers. A multiset Γ is called a *cluster* if the incompatibility graph $H(\Gamma)$ is connected. We also denote the set of all clusters from $\mathcal{P}$ as $\mathcal{C}$. Given $\Gamma \in \mathcal{C}$, let $H = H(\Gamma)$ and define the Ursell function

$$\Phi(\Gamma) := \frac{1}{|V(H)|!} \sum_{\substack{A \subseteq E(H) \\ \text{spanning, connected}}} (-1)^{|A|}. \tag{3}$$

▶ **Definition 6** (Cluster Expansion). *The cluster expansion of* Ξ *is*

$$\log \Xi(\mathcal{P}, w) := \sum_{\Gamma \in \mathcal{C}} \Phi(\Gamma) \prod_{\gamma \in \Gamma} w(\gamma).$$

We refer the interested reader to the survey by Jenssen [15] for various algorithmic (and beyond) applications of the cluster expansion. The cluster expansion is a good approximation to the logarithm of the partition function under various conditions. One of the most well-known conditions is the following.

[2] The weight function can be complex-valued, but in this paper we will focus on real-valued weight functions.

▶ **Proposition 7** (Kotecký-Preiss Criterion [17]). *Let $g : \mathcal{P} \to [0, \infty)$ be a "decay function". Suppose that for all $\gamma \in \mathcal{P}$, we have,*

$$\sum_{\gamma' \not\sim \gamma} |w(\gamma')| \, e^{|\gamma'|+g(\gamma')} \leq |\gamma| \,. \tag{4}$$

Then, the cluster expansion converges absolutely.

We note that the condition in Proposition 7 is not the most general, but this particular form will be convenient for our use later.

We are going to recast (1) into a polymer partition function. Let $G = (V, E)$ be a graph of maximum degree Δ. For $S \subseteq V$, let $\mathcal{K}(S)$ be the set of connected components in the induced subgraph $G[S]$.

Denote by $\overline{C_v}$ the constraint $\sum_{e \sim v} x_e > 1$. Using the indicator function for the constraint $\overline{C_v}$, and the fact that $\mathbb{1}_{C_v} = 1 - \mathbb{1}_{\overline{C_v}}$, we can expand (1) as

$$\begin{aligned}
\mathrm{Vol}(P_{G,\delta}) &= \int_{{M_\delta}^E} \prod_{v \in V} (1 - \mathbb{1}_{\overline{C_v}}) d\mu \\
&= \int_{{M_\delta}^E} \sum_{S \subseteq V} (-1)^{|S|} \prod_{v \in S} \mathbb{1}_{\overline{C_v}} d\mu \\
&= \int_{{M_\delta}^E} \sum_{S \subseteq V} \prod_{K \in \mathcal{K}(S)} (-1)^{|K|} \prod_{v \in K} \mathbb{1}_{\overline{C_v}} d\mu.
\end{aligned}$$

We then continue by pushing the integral inside:

$$\begin{aligned}
\mathrm{Vol}(P_{G,\delta}) &= \sum_{S \subseteq V} \int_{{M_\delta}^E} \prod_{K \in \mathcal{K}(S)} (-1)^{|K|} \prod_{v \in K} \mathbb{1}_{\overline{C_v}} d\mu \\
&= \sum_{S \subseteq V} \int_{{M_\delta}^{E \setminus E(S)}} 1 d\mu \left(\prod_{K \in \mathcal{K}(S)} (-1)^{|K|} \int_{{M_\delta}^{E(K)}} \prod_{v \in K} \mathbb{1}_{\overline{C_v}} d\mu \right),
\end{aligned} \tag{5}$$

where, for a subset $S \subseteq V$ of vertices, $E(S)$ denotes the set of edges adjacent to it. Note that we swapped the integral with the product in (5). This is valid because the constraint regarding each S_i rely on disjoint set of edges. There is also an extra factor of $\int_{{M_\delta}^{E \setminus E(S)}} 1 d\mu$ to account for the edges that are not adjacent to S. As $\int_{{M_\delta}^{E \setminus E(S)}} 1 d\mu = \left(\frac{1+\delta}{\Delta}\right)^{|E|-|E(S)|}$, we have

$$\begin{aligned}
\mathrm{Vol}(P_{G,\delta}) &= \left(\frac{1+\delta}{\Delta}\right)^{|E|} \sum_{S \subseteq V} \left(\frac{1+\delta}{\Delta}\right)^{-|E(S)|} \prod_{K \in \mathcal{K}(S)} \int_{{M_\delta}^{E(K)}} (-1)^{|K|} \prod_{v \in K} \mathbb{1}_{\overline{C_v}} d\mu \\
&= \left(\frac{1+\delta}{\Delta}\right)^{|E|} \sum_{S \subseteq V} \prod_{K \in \mathcal{K}(S)} w(K),
\end{aligned} \tag{6}$$

where

$$w(K) = \left(\frac{1+\delta}{\Delta}\right)^{-|E(K)|} (-1)^{|K|} \int_{{M_\delta}^{E(K)}} \prod_{v \in K} \mathbb{1}_{\overline{C_v}} d\mu. \tag{7}$$

The redistribution of the $\left(\frac{1+\delta}{\Delta}\right)^{-|E(S)|}$ factor is valid because $\{E(K)\}_{K \in \mathcal{K}(S)}$ is a partition of $E(S)$.

To fit (6) into Definition 5, we call a subset $S \subseteq V$ a polymer if the induced subgraph $G[S]$ is connected. Two polymers S_1 and S_2 are compatible if $\mathrm{dist}_G(S_1, S_2) \geq 2$, namely when they are not adjacent. The size of a polymer is simply the number of vertices it contains. In addition, equipped with the weight function in (7), we then have that

$$\Xi(G) = \frac{\mathrm{Vol}(P_{G,\delta})}{\left(\frac{1+\delta}{\Delta}\right)^{|E|}}. \tag{8}$$

The polymer model is useful thanks to the following theorem by Jenssen, Keevash, and Perkins [16] (building upon the works of [9, 20, 14]).

▶ **Proposition 8** ([16, Theorem 8]). *Fix an integer $\Delta > 0$ and let $\mathcal{G}$ be a class of graphs of maximum degree at most Δ. Suppose the following conditions hold for a given polymer model with decay function $g(\cdot)$ as in Proposition 7:*

1. *There exist constants $c_1, c_2 > 0$ such that given a connected subgraph γ, determining whether γ is a polymer, and then computing $w(\gamma)$ and $g(\gamma)$ can be done in time $O(|\gamma|^{c_1} e^{c_2|\gamma|})$;*
2. *There exists $\rho > 0$ such that for every $G \in \mathcal{G}$ and every polymer $\gamma \in \mathcal{P}(G)$, $g(\gamma) \geq \rho|\gamma|$;*
3. *The Kotecký-Preiss criterion as in Proposition 7 holds for $g(\cdot)$.*

Then there exists an FPTAS for $\Xi(G)$ for every $G \in \mathcal{G}$.

The algorithm in Proposition 8 is as follows. Given $\varepsilon > 0$ and a graph G on n vertices, let $t = \log(n/\varepsilon)$. For a cluster $\Gamma \in \mathcal{C}$, extend g by defining $g(\Gamma) := \sum_{\gamma \in \Gamma} g(\gamma)$. Moreover, recall the Ursell function in (3). Then,

1. Enumerate all clusters $\Gamma \in \mathcal{C}$ with $|\Gamma| < t/\rho$;
2. For each such Γ, compute $\Phi(\Gamma)$, $\prod_{\gamma \in \Gamma} w(\gamma)$, and $g(\Gamma)$;
3. Compute

$$T_t(G) := \sum_{\gamma \in \mathcal{C},\ |\Gamma| < t/\rho,\ g(\Gamma) < t} \Phi(\Gamma) \prod_{\gamma \in \Gamma} w(\gamma);$$

4. Output $\exp(T_t(G))$.

Roughly speaking, this yields a good approximation because the KP criterion in Proposition 7 holds. All the computation is efficient because of the algorithmic techniques in [9, 20, 14].

2.1 Verifying the Kotecký-Preiss Criterion

The most important condition in Proposition 8 is the Kotecký-Preiss criterion as in Proposition 7. Let $g(\gamma) := \rho\,|\gamma|$ for some sufficiently small constant ρ. For any $S \subseteq V$, let $N^+(S)$ be the extended neighbourhood of S, namely, $N^+(S) := S \cup \partial S$. Then we have,

$$\begin{aligned} \sum_{\gamma' \not\sim \gamma} |w(\gamma')|\, e^{|\gamma'| + g(\gamma')} &= \sum_{\gamma' \not\sim \gamma} |w(\gamma')|\, e^{(1+\rho)|\gamma'|} \\ &\leq \sum_{v \in N^+(\gamma)} \sum_{\gamma' \ni v} |w(\gamma')|\, e^{(1+\rho)|\gamma'|}. \end{aligned} \tag{9}$$

Note that $|N^+(\gamma)| \leq (\Delta + 1)\,|\gamma|$. Moreover, Borgs, Chayes, Kahn, and Lovász showed the following lemma.

▶ **Lemma 9** ([10, Lemma 2.1]). *For a graph G of maximum degree Δ, the number of connected subgraphs containing v of size ℓ is at most $\frac{1}{\ell}\binom{\ell\Delta}{\ell-1} \leq (e\Delta)^{\ell-1}$.*

Thus we need an upper bound of $w(\gamma)$ for any polymer γ of a fixed size ℓ.

▶ **Lemma 10.** *For a polymer γ of size ℓ and $\delta < \frac{1}{\Delta-1}$,*

$$|w(\gamma)| \le (e\delta)^{|\gamma|}.$$

Proof. Note that we can reinterpret (7) as

$$|w(\gamma)| = \Pr\left[\bigwedge_{v\in\gamma} \overline{C_v}\right],$$

where the probability is over the product distribution of $\{x_e\}_{e\in E(\gamma)}$ where each x_e is uniform over $\left[0, \frac{1+\delta}{\Delta}\right]$. Let I_γ be a maximal independent set of γ. Then $|I_\gamma| \ge \frac{|\gamma|}{\Delta}$, and

$$\Pr\left[\bigwedge_{v\in\gamma} \overline{C_v}\right] \le \Pr\left[\bigwedge_{v\in I_\gamma} \overline{C_v}\right] = \prod_{v\in I_\gamma} \Pr\left[\,\overline{C_v}\,\right]. \tag{10}$$

Recall that $\overline{C_v}$ is the constraint that $\sum_{e\sim v} x_e > 1$.

If there is some $u \in I_\gamma$ such that $\deg_G(u) \le \Delta - 1$, then $\sum_{e\sim u} x_e \le \frac{\Delta-1}{\Delta}(1+\delta) < 1$ as $\delta < \frac{1}{\Delta-1}$ In this case $\Pr\left[\,\overline{C_u}\,\right] = 0$ and the lemma follows.

Therefore we can assume that $\deg_G(v) = \Delta$ for all $v \in I_\gamma$. Let $y_e = \frac{1+\delta}{\Delta} - x_e$. Then

$$\Pr\left[\sum_{e\sim v} x_e > 1\right] = \Pr\left[\sum_{e\sim v} y_e < \delta\right],$$

where each y_e is uniform over $\left[0, \frac{1+\delta}{\Delta}\right]$.
Denote by S_δ the simplex $\{\mathbf{y} \mid \sum_{e\sim v} y_e < \delta, \text{ and } \forall e \sim v,\ y_e \ge 0\}$. Then, $S_\delta \subseteq {M_\delta}^{E(v)}$ as $\delta < \frac{1+\delta}{\Delta}$. Thus,

$$\Pr\left[\sum_{e\sim v} y_e < \delta\right] = \frac{\mathrm{Vol}\left(S_\delta \cap {M_\delta}^{E(v)}\right)}{\mathrm{Vol}\left({M_\delta}^{E(v)}\right)} = \frac{\mathrm{Vol}\left(S_\delta\right)}{\mathrm{Vol}\left({M_\delta}^{E(v)}\right)} = \frac{\frac{\delta^\Delta}{\Delta!}}{\left(\frac{1+\delta}{\Delta}\right)^\Delta} \le \left(\frac{e\delta}{1+\delta}\right)^\Delta,$$

where we used the fact that a standard simplex of dimension Δ has volume $1/\Delta!$ and $\Delta! \ge \left(\frac{\Delta}{e}\right)^\Delta$. Plugging it back to (10),

$$|w(\gamma)| \le \prod_{v\in I_\gamma} \Pr\left[\,\overline{C_v}\,\right] \le \left(\frac{e\delta}{1+\delta}\right)^{\Delta\cdot\frac{|\gamma|}{\Delta}} < (e\delta)^{|\gamma|}. \qquad ◀$$

Now we can verify the Kotecký-Preiss criterion.

▶ **Lemma 11.** *For graphs of maximum degree $\Delta \ge 2$, $\delta \le \frac{1}{e^4\Delta}$, and sufficiently small ρ, the Kotecký-Preiss criterion (4) holds.*

Proof. By (9), Lemma 9, and Lemma 10, for sufficiently small ρ,

$$\begin{aligned}
\sum_{\gamma\not\sim\gamma'} |w(\gamma')|\, e^{|\gamma'|+g(\gamma')} &\le (\Delta+1)\,|\gamma| \sum_{\ell\ge1} (e\Delta)^{\ell-1}\,(e\delta)^\ell\, e^{(1+\rho)\ell} \\
&\le \frac{\Delta+1}{e\Delta}\,|\gamma| \sum_{\ell\ge1} \left(e^{3+\rho}\delta\Delta\right)^\ell \le \frac{1.5}{e}\,|\gamma| \sum_{\ell\ge1} \left(e^{\rho-1}\right)^\ell \\
&\qquad\qquad \text{(as } \delta \le \tfrac{1}{e^4\Delta} \text{ and } \Delta \ge 2) \\
&\le \left(\frac{1.5}{e}\cdot\frac{e^{\rho-1}}{1-e^{\rho-1}}\right)|\gamma| < |\gamma|. \qquad ◀
\end{aligned}$$

2.2 Proof of Theorem 3

With the KP criterion verified, we can prove Theorem 3 now.

Proof of Theorem 3. We apply Proposition 8. Let $g(\gamma) = \rho|\gamma|$ for a sufficiently small constant ρ, which then satisfies Item (2). The Item (3) follows from Lemma 11. For Item (1), determining if γ is a polymer is trivial, and computing $g(\gamma)$ is trivial too.

For computing $w(\gamma)$, recall (7). It is easy to see that $\left(\frac{1+\delta}{\Delta}\right)^{|E(\gamma)|}(-1)^{|\gamma|}w(\gamma)$ is in fact the volume of a polytope with variables $\{x_e\}$ for each $e \in E(\gamma)$, and constraints $\overline{C_v}$ for each $v \in \gamma$ and $0 \le x_e \le \frac{1+\delta}{\Delta}$ for each $e \in E(\gamma)$. This polytope has at most $\binom{|E(\gamma)|+|\gamma|}{|E(\gamma)|} = \binom{|E(\gamma)|+|\gamma|}{|\gamma|} = O((e(\Delta+1))^{|\gamma|})$ vertices, as $|E(\gamma)| \le \Delta|\gamma|$. Using the polytope volume algorithm by Lawrence [18] or by Barvinok [5], we can thus compute $w(\gamma)$ in $O(e^{C|\gamma|})$ time for some constant C that depends on Δ. ◀

▶ Remark 12. The bit complexity of the volume of a polytope can be exponential in the size of the polytope. Recall the algorithm (described after Proposition 8) of [16]. The polytopes for which we apply the algorithm of [18] or [5] essentially have size $O(\log n)$. Thus the bit complexity of their volumes is at most a polynomial in n.

Moreover, Lawrence's algorithm [18] is only stated for simple polytopes. To get around this issue, one may employ a small perturbation, such as the one from [19], to make the polytope simple. Doing so incurs an exponentially small additive error on the volume. On the other hand, Barvinok's algorithm [5] does not require simple polytopes, but it also incurs exponentially small additive errors. It is easy to observe that, as these volumes are in the exponent of the final estimate, small additive errors translate to small multiplicative errors in the end. Thus, for input error tolerance ε, we can set $\varepsilon' = \varepsilon/2$ in the algorithm of [16] to accommodate the extra small error in volume computation.

3 Hypergraph Matching Polytope

Now we turn our attention to the hypergraph case and show Theorem 4. Let $H = (V, E)$ be a hypergraph. We will assume throughout this section that, for some constants $\Delta, k \ge 2$, vertices of H have maximum degree Δ, and hyperedges of H have maximum size k. Recall that the fractional hypergraph matching polytope is defined in (2), and the truncated version $P_{H,\delta}$ is defined by intersecting P_H with ${M_\delta}^E$.

Our δ will satisfy that $\delta < \frac{1}{\Delta-1}$, in which case, if the degree of some v is at most $\Delta - 1$, the corresponding constraint is always satisfied. Thus, we may assume that $\deg_H(v) = \Delta$ for all $v \in V$.

Let $\mathrm{Flat}(H)$ be the flattened graph of H, namely its vertex set is still V, and for $u, v \in V$, u and v are adjacent if they appear in the same hyperedge e for some $e \in E$. In other words, we replace each hyperedge e by a clique on the vertex set of e. Notice that the maximum degree of $\mathrm{Flat}(H)$ is at most $\Delta(k-1)$. Similar to (5), we have

$$\mathrm{Vol}(P_{H,\delta}) = \sum_{S \subseteq V} \int_{{M_\delta}^{E \setminus E(S)}} 1 d\mu \left(\prod_{K \in \mathcal{K}(S)} (-1)^{|K|} \int_{{M_\delta}^{E(K)}} \prod_{v \in K} \mathbb{1}_{\overline{C_v}} d\mu \right), \tag{11}$$

where $\mathcal{K}(S)$ denote the connected components of the induced subgraph $\mathrm{Flat}(H)[S]$. Thus, we have a similar polymer model partition function

$$\Xi(H) = \frac{\mathrm{Vol}(P_{H,\delta})}{\left(\frac{1+\delta}{\Delta}\right)^{|E|}} = \sum_{S \subseteq V} \prod_{K \in \mathcal{K}(S)} w_H(K), \tag{12}$$

where

$$w_H(K) = \left(\frac{1+\delta}{\Delta}\right)^{-|E(K)|} (-1)^{|K|} \int_{M_\delta{}^{E(K)}} \prod_{v\in K} \mathbb{1}_{\overline{C_v}} d\mu. \tag{13}$$

While (12) and (13) look identical to (6) and (7), the underlying graph is different and consequently Lemma 10 no longer applies. We may view this as a polymer model over $\mathrm{Flat}(H)$, by treating connected induced subgraphs of $\mathrm{Flat}(H)$ as polymers.

Directly applying the method of Lemma 10 would yield

$$|w_H(\gamma)| < (e\delta)^{\frac{|\gamma|}{k-1}}.$$

This bound can only validate the Kotecký-Preiss criterion up to $\delta = \Theta\left(\frac{1}{\Delta k}\right)^{k-1}$. To get a better bound, we need a different polymer model.

3.1 A Different Polymer Model

Let $\mathrm{Inc}(H)$ be the incidence graph of H, where the vertex set is $V \cup E$, and for $v \in V$ and $e \in E$, $v \sim e$ in $\mathrm{Inc}(H)$ if $v \in e$ in H. Then, $\mathrm{Inc}(H)$ is a bipartite graph, and $\mathrm{Flat}(H)$ is exactly $\mathrm{Inc}(H)$ projected on the V side. For each subset $S \subset V$ of vertices, let $\partial_{\mathrm{Inc}(H)} S$ be the neighbours of S in $\mathrm{Inc}(H)$. Note that $\partial_{\mathrm{Inc}(H)} S = E(S)$, namely the set of hyperedges that contain at least one vertex in S.

We rewrite (12) into a different polymer model. For a connected induced subgraph K of $\mathrm{Flat}(H)$, we map it to a *minimal connected subgraph* (MCS) T as follows. Fix an arbitrary ordering of V. We consider vertices of K in order one at a time, to potentially add them to T. Initialise T to be the first vertex of K and remove it from K. Given the current T, we find the first vertex u in K that is adjacent to the current T in $\mathrm{Flat}(H)$. Add u to T if $E(v_i) \not\subseteq E(T)$. In other words, we only add v_i if it introduces a new hyperedge to $E(T)$. Whether u is added to T or not, we remove u from K. We keep doing this until K is empty. Denote this mapping by φ. We also extend the mapping to any (not necessarily connected) subset $S \subseteq V$ by defining $\varphi(S) = \emptyset$ if $\mathrm{Flat}(H)[S]$ is not connected. We call a non-empty image T of φ an MCS, namely, T is an MCS if there is a connected $K \subseteq V$ such that $\varphi(K) = T$. Equivalently, a connected T is an MCS if and only if $\varphi(T) = T$.

▶ **Lemma 13.** *Let T be an MCS. Then,*
1. $|E(T)| \geq |T|$*;*
2. T *is connected in* $\mathrm{Flat}(H)$*.*

Proof. Both claims are straightforward by an induction on the number of vertices of T. ◀

While an MCS is connected in $\mathrm{Flat}(H)$, it is not necessarily a tree. Indeed, an MCS is determined by $\mathrm{Inc}(H)$ and not by $\mathrm{Flat}(H)$ alone.

Our new model have MCSes as polymers. As before two MCSes are incompatible if their distance is at most 1. For this model, define a new weight function

$$u(T) := \sum_{K:\varphi(K)=T} w_H(K).$$

We rewrite (12) as

$$\Xi(H) = \sum_F \prod_{T\in\mathcal{K}(F)} u(F), \tag{14}$$

where the sum is over $F \subseteq V$ whose every component is an MCS.

For an MCS T, a vertex $v \in V \setminus T$ is said to be *broken* if $\varphi(T \cup \{v\}) = T$. Let $B(T)$ be the set of broken vertices with respect to T. This notion is a hypergraph generalisation of the well-known broken circuit theory for graphs [23, 21].

▶ **Lemma 14.** *For $K \subseteq V$, $\varphi(K) = T$ if and only if $T \subseteq K \subseteq (T \cup B(T))$.*

Proof. If $\varphi(K) = T$, the inclusion $T \subseteq K$ follows directly from the definition of φ. For the other inclusion, we induct on the size of K. If $K = T$, the claim holds trivially. For the inductive step, let $v \in K \setminus T$ be the first vertex that is not added to T during φ. Then, clearly $\varphi(K \setminus \{v\}) = T$, and thus by the induction hypothesis $K \setminus \{v\} \subseteq (T \cup B(T))$. We claim that $v \in B(T)$ as well. This is because the mapping $\varphi(T \cup \{v\})$ would behave the same as $\varphi(K)$ until v is considered (and then dropped), as before v all vertices considered in $\varphi(K)$ are added to T. As v is dropped next, the rest of the process $\varphi(T \cup \{v\})$ just processes vertices of T and add them. It implies that $\varphi(T \cup \{v\}) = T$, and thus $v \in B(T)$. This finishes the inductive step and shows that $K \subseteq (T \cup B(T))$.

For the other direction, suppose $T \subseteq K \subseteq (T \cup B(T))$. We also do an induction on $|K|$. The base case of $K = T$ is trivial. For the inductive step, consider the first $v \in K \setminus T$ that is processed during $\varphi(K)$. Note that up to v all processed vertices are in T, and the process is identical to the process of $\varphi(T \cup \{v\})$ up to this point. Thus, v would be discarded then. The rest of $\varphi(K)$ is identical to $\varphi(K \setminus \{v\})$, and by the induction hypothesis, $\varphi(K \setminus \{v\}) = T$. Thus, $\varphi(K) = T$. ◀

Notice that if $\varphi(K) = T$, $E(K) = E(T)$. We then have a better expression for $u(T)$ as follows,

$$u(T) = \sum_{K:\varphi(K)=T} w_H(K) = \sum_{K:\varphi(K)=T} \left(\frac{1+\delta}{\Delta}\right)^{-|E(T)|} (-1)^{|K|} \int_{{M_\delta}^{E(K)}} \prod_{v\in K} \mathbb{1}_{\overline{C_v}} d\mu \qquad \text{(by (13))}$$

$$= \left(\frac{1+\delta}{\Delta}\right)^{-|E(T)|} \sum_{K:T\subseteq K\subseteq T\cup B(T)} (-1)^{|K|} \int_{{M_\delta}^{E(T)}} \prod_{v\in K} \mathbb{1}_{\overline{C_v}} d\mu \qquad \text{(by Lemma 14)}$$

$$= \left(\frac{1+\delta}{\Delta}\right)^{-|E(T)|} (-1)^{|T|} \sum_{A\subseteq B(T)} \int_{{M_\delta}^{E(T)}} \prod_{v\in T} \mathbb{1}_{\overline{C_v}} \prod_{v\in A} -\mathbb{1}_{\overline{C_v}} d\mu$$

$$= \left(\frac{1+\delta}{\Delta}\right)^{-|E(T)|} (-1)^{|T|} \int_{{M_\delta}^{E(T)}} \prod_{v\in T} \mathbb{1}_{\overline{C_v}} \prod_{v\in B(T)} \left(1 - \mathbb{1}_{\overline{C_v}}\right) d\mu$$

$$= \left(\frac{1+\delta}{\Delta}\right)^{-|E(T)|} (-1)^{|T|} \int_{{M_\delta}^{E(T)}} \prod_{v\in T} \mathbb{1}_{\overline{C_v}} \prod_{v\in B(T)} \mathbb{1}_{C_v} d\mu. \qquad (15)$$

Next we give an upper bound for $|u(T)|$.

▶ **Lemma 15.** *Let T be a MCS of size ℓ. For $\delta < \frac{1}{\Delta}$,*

$$|u(T)| \leq (\Delta\delta)^\ell \left(\frac{e}{\Delta}\right)^{\frac{\ell}{k-1}}.$$

Proof. By (15),

$$|u(T)| \leq \left(\frac{1+\delta}{\Delta}\right)^{-|E(T)|} \int_{{M_\delta}^{E(T)}} \prod_{v\in T} \mathbb{1}_{\overline{C_v}} d\mu.$$

Thus,

$$|u(T)| \leq \Pr\left[\bigwedge_{v\in T} \overline{C_v}\right],$$

where the probability is over the product distribution of x_e's for $e \in E(T)$, and each x_e is uniform over $\left[0, \frac{1+\delta}{\Delta}\right]$. Notice that $\overline{C_v}$ means $\sum_{e\sim v} x_e > 1$, which is equivalent to $\sum_{e\sim v} y_e < \delta$ where $y_e = \frac{1+\delta}{\Delta} - x_e$. As $y_e \geq 0$, this implies that $y_e < \delta$ for all $e \sim v$. Since $T \cup E(T)$ is connected in $\mathrm{Inc}(H)$, if $\prod_{v\in T} \mathbb{1}_{\overline{C_v}} = 1$, then for all $e \in E(T)$, $y_e < \delta$.

Moreover, for a maximal independent set I_T of T (in $\mathrm{Flat}(H)$), we have that $|I_T| \geq \frac{|T|}{\Delta(k-1)}$, as the maximum degree of $\mathrm{Flat}(H)$ is $\Delta(k-1)$. Note that for $v \in I_T$, the constraints C_v are independent from each other. Then we have

$$\begin{aligned}
\Pr\left[\bigwedge_{v\in T} \overline{C_v}\right] &\leq \Pr\left[\bigwedge_{v\in I_T} \overline{C_v} \wedge \bigwedge_{e\in E(T)\setminus E(I_T)} y_e < \delta\right] \\
&= \prod_{v\in I_T} \Pr\left[\,\overline{C_v}\,\right] \prod_{e\in E(T)\setminus E(I_T)} \Pr\left[y_e < \delta\right] \\
&= \left(\frac{1+\delta}{\Delta}\right)^{-|E(T)|} \delta^{|E(T)|-|E(I_T)|} \left(\frac{\delta^\Delta}{\Delta!}\right)^{|I_T|},
\end{aligned}$$

where we use the fact that the volume of a standard simplex of dimension Δ is $\frac{1}{\Delta!}$. (Recall that we can assume that all vertices have degree Δ, as vertices of smaller degrees correspond to a constraint that trivially holds.) As I_T is an independent set, $|E(I_T)| = \Delta\,|I_T|$. Together with $\Delta! \geq \left(\frac{\Delta}{e}\right)^\Delta$ and $|I_T| \geq \frac{|T|}{\Delta(k-1)}$, we have

$$\Pr\left[\bigwedge_{v\in T} \overline{C_v}\right] \leq (\Delta\delta)^{|E(T)|} \left(\left(\frac{e}{\Delta}\right)^\Delta\right)^{\frac{|T|}{\Delta(k-1)}}.$$

As T is a MCS, by Lemma 13, $|E(T)| = \left|\partial_{\mathrm{Inc}(H)} T\right| \geq |T| = \ell$. Then, as $\delta\Delta < 1$,

$$|u(T)| \leq (\Delta\delta)^\ell \left(\frac{e}{\Delta}\right)^{\frac{\ell}{k-1}}. \qquad \blacktriangleleft$$

Given Lemma 15, the verification of the Kotecký-Preiss criterion (4) is very similar to previous arguments.

▶ **Lemma 16.** *For hypergraphs of maximum degree $\Delta \geq 2$ and maximum hyperedge size $k \geq 2$, $\delta \leq \left(e^4 \Delta^{\frac{2k-3}{k-1}} (k-1)\right)^{-1}$, and sufficiently small ρ, the Kotecký-Preiss criterion (4) holds for MCSes and $u(\cdot)$.*

Proof. Let T be a MCS. Any MCS incompatible with T must contain at least a vertex of T or a neighbour of T in $\mathrm{Flat}(H)$. By Lemma 13, every MCS is a connected subgraph in $\mathrm{Flat}(H)$. For any $v \in T \cup \partial_{\mathrm{Flat}(H)} T$, by Lemma 9, the number of MCSes of size ℓ that contains v is at most $(e\Delta(k-1))^{\ell-1}$. As $\left|T \cup \partial_{\mathrm{Flat}(H)} T\right| \leq (\Delta(k-1)+1)\,|T|$, the number of MCSes of size ℓ and incompatible with T is at most $(\Delta(k-1)+1)\,|T|\,(e\Delta(k-1))^{\ell-1}$. Using this bound, for $\Delta \geq 2$, $k \geq 2$, and sufficiently small constant ρ,

$$\begin{aligned}\sum_{T' \not\sim T} |w(T')|\, e^{|T'|+g(T')} &\le (\Delta(k-1)+1)\,|T| \sum_{\ell \ge 1} (e\Delta(k-1))^{\ell-1} (\Delta\delta)^{\ell} \left(\frac{e}{\Delta}\right)^{\frac{\ell}{k-1}} e^{(1+\rho)\ell} \\ &= \frac{\Delta(k-1)+1}{e\Delta(k-1)}\,|T| \sum_{\ell \ge 1} \left(e^{2+\frac{1}{k-1}+\rho} \delta \Delta^{\frac{2k-3}{k-1}} (k-1)\right)^{\ell} \\ &\le \frac{1.5}{e}\,|T| \sum_{\ell \ge 1} \left(e^{\rho-1}\right)^{\ell} \\ &\le \left(\frac{1.5}{e} \cdot \frac{e^{\rho-1}}{1-e^{\rho-1}}\right) |T| < |T|\,. \end{aligned}$$

◀

3.2 Proof of Theorem 4

With the KP criterion verified, we can prove Theorem 4 now.

Proof of Theorem 4. We apply Proposition 8 on $\mathrm{Flat}(H)$. For a polymer (MCS) T, let $g(T) = \rho\,|T|$ for a sufficiently small constant T, which then satisfies Item (2). Item (3) follows from Lemma 16. For Item (1), computing $g(T)$ is trivial. To determine if T is a polymer is trivial, we just check if $\varphi(T) = T$. This takes time linear in $|T|$.

For computing $u(T)$, we use the expression (15). The integral in (15) is the volume of a polytope defined by the constraints $\overline{C_v}$ for $v \in T$, C_v for $v \in B(T)$, and $0 \le x_e \le \frac{1+\delta}{\Delta}$ for $e \in E(T)$. To find the defining constraints of this polytope, we need to compute $B(T)$. Note that by definition, $v \in B(T)$ only if $T \cup \{v\}$ is connected, which implies that $B(T) \subseteq \partial_{\mathrm{Flat}(H)} T$. Then we just need to go through every $v \in \partial_{\mathrm{Flat}(H)} T$ and check if $\varphi(T \cup \{v\}) = T$. As $\left|\partial_{\mathrm{Flat}(H)} T\right| \le \Delta(k-1)\,|T|$, this takes at most $O(|T|^2)$ time.

There are $|T| + |B(T)| + |E(T)|$ constraints and $|E(T)|$ variables. Note that $|B(T)| \le \left|\partial_{\mathrm{Flat}(H)} T\right| \le \Delta(k-1)\,|T|$. Recall that $|E(T)| \le \Delta\,|T|$ and by Lemma 13, $|E(T)| \ge |T|$. Thus, the number of vertices of this polytope is at most

$$\begin{aligned}\binom{|T|+|B(T)|+|E(T)|}{|E(T)|} &\le \left(\frac{e(|T|+|B(T)|+|E(T)|)}{|E(T)|}\right)^{|E(T)|} \\ &\le (e\Delta k + e)^{\Delta|T|}.\end{aligned}$$

Using the polytope volume algorithm by Lawrence [18] or by Barvinok [5], we can compute $u(T)$ in $O(e^{C|\gamma|})$ time for some constant C that depends on Δ and k. This verifies Item (1) of Proposition 8 and finishes the proof. ◀

Regarding the applicability of [18] and [5], see Remark 12.

4 Concluding Remarks

Our work leaves open the question of whether, under the setting of Theorem 4, an FPTAS exists for $\delta = \frac{C}{\Delta k}$ for some constant $C > 0$. The most straightforward potential approach to this bound is to strengthen the bound in Lemma 15 to $(C\delta)^{|T|}$. However, this strengthening does not appear to hold, at least for some parameters k and Δ, when $\delta = \frac{C}{\Delta k}$. To see this, consider $T = \{v_1, \ldots, v_k\}$ of size k. Let the hyperedges $e_1 = \cdots = e_{\Delta-1} = T$. Moreover, for each $i \in [k]$, introduce f_i such that $v_i \in f_i$ but $v_j \notin f_i$ for any $j \ne i$. Namely, each f_i contains only v_i from T (and possibly other vertices from outside of T). Note that T is an MCS and let $B(T) = \emptyset$. Then, $|T| = k$ and $|E(T)| = \Delta - 1 + k$. We have

$$\begin{aligned}\int_{M_\delta{}^{E(T)}} \prod_{v\in T} \mathbb{1}_{\overline{C_v}} d\mu &= \sum_{i=1}^{k} \int_{M_\delta{}^{E(T)}} \mathbb{1}_{\overline{C_{v_i}}} \prod_{j=1}^{k} \mathbb{1}_{x_{f_j}\geq x_{f_i}} d\mu \\ &= \sum_{i=1}^{k} \int_{M_\delta{}^{E(T)}} \mathbb{1}_{\sum_{t=1}^{\Delta-1} y_{e_t}\leq \delta - y_{f_i}} \prod_{j=1}^{k} \mathbb{1}_{y_{f_j}\leq y_{f_i}} d\mu \\ &= \sum_{i=1}^{k} \int_0^\delta \frac{(\delta - y_{f_i})^{\Delta-1} y_{f_i}^{k-1}}{(\Delta-1)!} dy_{f_i} \\ &= \frac{k!\delta^{\Delta-1+k}}{(\Delta-1+k)!},\end{aligned}$$

where we substitute $y_e = \frac{1+\delta}{\Delta} - x_e$ for all $e \in \{e_1, \ldots, e_{\Delta-1}, f_1, \ldots, f_k\}$. Then,

$$\begin{aligned}|u(T)| &= \left(\frac{1+\delta}{\Delta}\right)^{-(\Delta-1+k)} \frac{k!\delta^{\Delta-1+k}}{(\Delta-1+k)!} \\ &\geq \left(\frac{1+\delta}{\Delta}\right)^{-(\Delta-1+k)} \frac{\delta^{\Delta-1+k}}{(\Delta-1+k)^{\Delta-1}} \\ &= (1+\delta)^{-(\Delta-1+k)} \left(\frac{\Delta\delta}{\Delta-1+k}\right)^{\Delta-1} (\Delta\delta)^k.\end{aligned}$$

Plug in $\delta = \frac{C}{\Delta k}$ and let $k \geq \Delta$ such that $\frac{k}{\log k} \gg \frac{\Delta}{\log \Delta}$ (for example $k = \Delta^2$). Then,

$$\begin{aligned}\frac{|u(T)|}{\delta^k} &\geq C_0 \left(\frac{C}{2k^2}\right)^{\Delta-1} \Delta^k \\ &\geq C_1^k = C_1^{|T|},\end{aligned}$$

for some constant $C_0 > 0$ and any constant $C_1 > 0$ when $\Delta \to \infty$. Thus, to achieve an FPTAS for $\delta = \frac{C}{\Delta k}$, some new ideas are required.

References

1. Antar Bandyopadhyay and David Gamarnik. Counting without sampling: new algorithms for enumeration problems using statistical physics. In *SODA*, pages 890–899. ACM, 2006. URL: http://dl.acm.org/citation.cfm?id=1109557.1109655.
2. Imre Bárány and Zoltán Füredi. Computing the volume is difficult. *Discrete Comput. Geom.*, 2(4):319–326, 1987. doi:10.1007/BF02187886.
3. Alexander Barvinok. Asymptotic estimates for the number of contingency tables, integer flows, and volumes of transportation polytopes. *Int. Math. Res. Not.*, 2009(2):348–385, 2009.
4. Alexander Barvinok and Mark Rudelson. A quick estimate for the volume of a polyhedron. *arXiv*, 2021. arXiv:2112.06322.
5. Alexander I. Barvinok. Computing the volume, counting integral points, and exponential sums. *Discrete Comput. Geom.*, 10(2):123–141, 1993. doi:10.1007/BF02573970.
6. Alexander I. Barvinok. *Combinatorics and Complexity of Partition Functions*, volume 30 of *Algorithms and combinatorics*. Springer, 2016. doi:10.1007/978-3-319-51829-9.
7. Mohsen Bayati, David Gamarnik, Dimitriy A. Katz, Chandra Nair, and Prasad Tetali. Simple deterministic approximation algorithms for counting matchings. In *STOC*, pages 122–127. ACM, 2007. doi:10.1145/1250790.1250809.
8. Ferenc Bencs and Guus Regts. Approximating the volume of a truncated relaxation of the independence polytope. *arXiv*, 2024. doi:10.48550/arXiv.2404.08577.

9 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Computing the Tutte polynomial in vertex-exponential time. In *FOCS*, pages 677–686. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.40.

10 Christian Borgs, Jennifer T. Chayes, Jeff Kahn, and László Lovász. Left and right convergence of graphs with bounded degree. *Random Struct. Algorithms*, 42(1):1–28, 2013. doi:10.1002/RSA.20414.

11 Martin E. Dyer, Alan M. Frieze, and Ravi Kannan. A random polynomial time algorithm for approximating the volume of convex bodies. *J. ACM*, 38(1):1–17, 1991. doi:10.1145/102782.102783.

12 György Elekes. A geometric inequality and the complexity of computing volume. *Discrete Comput. Geom.*, 1(4):289–292, 1986. doi:10.1007/BF02187701.

13 David Gamarnik and Devin Smedira. Computing the volume of a restricted independent set polytope deterministically. *arXiv*, 2023. doi:10.48550/arXiv.2312.03906.

14 Tyler Helmuth, Will Perkins, and Guus Regts. Algorithmic Pirogov-Sinai theory. *Probab. Theory Related Fields*, 176(3-4):851–895, 2020.

15 Matthew Jenssen. The cluster expansion in combinatorics. In *Surveys in combinatorics 2024*, volume 493 of *London Math. Soc. Lecture Note Ser.*, pages 55–88. Cambridge Univ. Press, Cambridge, 2024.

16 Matthew Jenssen, Peter Keevash, and Will Perkins. Algorithms for #BIS-hard problems on expander graphs. *SIAM J. Comput.*, 49(4):681–710, 2020. doi:10.1137/19M1286669.

17 Roman Kotecký and David Preiss. Cluster expansion for abstract polymer models. *Comm. Math. Phys.*, 103(3):491–498, 1986.

18 Jim Lawrence. Polytope volume computation. *Math. Comp.*, 57(195):259–271, 1991.

19 Nimrod Megiddo and Ramaswamy Chandrasekaran. On the ϵ-perturbation method for avoiding degeneracy. *Oper. Res. Lett.*, 8(6):305–308, 1989.

20 Viresh Patel and Guus Regts. Deterministic polynomial-time approximation algorithms for partition functions and graph polynomials. *SIAM J. Comput.*, 46(6):1893–1919, 2017. doi:10.1137/16M1101003.

21 William T. Tutte. A contribution to the theory of chromatic polynomials. *Canad. J. Math.*, 6:80–91, 1954.

22 Dror Weitz. Counting independent sets up to the tree threshold. In *STOC*, pages 140–149. ACM, 2006. doi:10.1145/1132516.1132538.

23 Hassler Whitney. A logical expansion in mathematics. *Bull. Amer. Math. Soc.*, 38(8):572–579, 1932.

The Computational Complexity of Factored Graphs

Shreya Gupta
University of California San Diego, La Jolla, CA, USA

Boyang Huang
University of California San Diego, La Jolla, CA, USA

Russell Impagliazzo
University of California San Diego, La Jolla, CA, USA

Stanley Woo
University of California San Diego, La Jolla, CA, USA

Christopher Ye
University of California San Diego, La Jolla, CA, USA

Abstract

While graphs and abstract data structures can be large and complex, practical instances are often regular or highly structured. If the instance has sufficient structure, we might hope to compress the object into a more succinct representation. An efficient algorithm (with respect to the compressed input size) could then lead to more efficient computations than algorithms taking the explicit, uncompressed object as input. This leads to a natural question: when does knowing the input instance has a more succinct representation make computation easier?

We initiate the study of the computational complexity of problems on factored graphs: graphs that are given as a formula of products and unions on smaller graphs. For any graph problem, we define a parameterized version that takes factored graphs as input, parameterized by the number of (smaller) ordinary graphs used to construct the factored graph. In this setting, we characterize the parameterized complexity of several natural graph problems, exhibiting a variety of complexities. We show that a decision version of lexicographically first maximal independent set is **XP**-complete, and therefore *unconditionally* not fixed-parameter tractable (**FPT**). On the other hand, we show that clique counting is **FPT**. Finally, we show that reachability is **XNL**-complete. Moreover, **XNL** is contained in **FPT** if and only if **NL** is contained in some fixed polynomial time.

2012 ACM Subject Classification Theory of computation → Complexity theory and logic; Theory of computation → Complexity classes; Theory of computation → Problems, reductions and completeness

Keywords and phrases Parameterized Complexity, Fine-grained complexity, Fixed-parameter tractability, Graph algorithms

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.58

Related Version *Full Version*: `https://arxiv.org/abs/2407.19102` [37]

Funding *Russell Impagliazzo*: Supported by NSF Award AF: Medium 2212136.
Christopher Ye: Supported by NSF Award AF: Medium 2212136, NSF grants 1652303, 1909046, 2112533, and HDR TRIPODS Phase II grant 2217058.

Acknowledgements We would like to thank Antonina Kolokolova, Anthony Ostuni, and anonymous reviewers for their many helpful comments and suggestions.

1 Introduction

Traditionally, algorithm design and computational complexity both measure computational time as a function of the input size. Thus, the complexity of computational problems is crucially sensitive to the way the instances are represented as bit sequences. While graphs and abstract data structures can be complex and expressive in the worst case, practical instances


© Shreya Gupta, Boyang Huang, Russell Impagliazzo, Stanley Woo, and Christopher Ye;
licensed under Creative Commons License CC-BY 4.0



are often highly structured or regular. For example, road networks are often organized into repetitive grid patterns. Similarly, in databases, relations frequently inherit underlying structures from previous relations through operations like joins. In molecular geometry, compounds such as graphite are composed of layers of graphene, with each layer forming a regular honeycomb structure. If the input instance has sufficiently regular structure, we might hope to compress the object into a more succinct representation.

An algorithm that is efficient with respect to the size of the succinct encodings would then be able to compute the desired result more efficiently than an algorithm that takes a naive representation of the input. This raises the question: when does knowing that the input instance was created in a uniform way (or has a succinct representation) make computation easier?

To address this question, various formulations of "succinct representation" have been explored. For example, one such formulation is given by a Boolean circuit that can produce any particular input bit [35, 61]. Frequently, the complexity of problems given in this type of succinct format is exponentially more difficult than when the input is given explicitly (see e.g. [35, 31, 49, 48]). However, it's important to note that the difficulty only increases as a function of the smaller size parameter, and the actual problem has not necessarily become more difficult.

Another type of succinct representation, factored instances, was introduced by [21]. While they give a number of specific problems rather than introduce the concept abstractly, we can generalize by thinking of a fixed set of operations that take pairs of instances to possibly larger instances. For example, one natural operation might be set sum, taking two sets of integers A, B to the set of all sums $A + B = \{a + b \mid a \in A, b \in B\}$. Instead of being described directly, the input to a factored problem is given as this operation applied to a pair of possibly smaller instances. If these operations can be computed in polynomial time, the size of the output might be polynomially larger than the input, so this representation might be considerably smaller than the original. For each such set of operations and problem on instances, we can define a parameterized version of the problem on factored instances, where the input is represented as a formula in these (binary) operations over smaller instances, with the parameter k being the number of smaller instances. If the underlying problem is polynomial time solvable, then for fixed k, the problem remains polynomial time solvable, but with an exponent that potentially grows with k. How the compressed representation affects the difficulty of the problems can be formalized in terms of the complexity of such factored problems from the point of view of parameterized complexity. In this paper, we consider such parameterized factored problems for graphs, using standard graph products and union as our operations.

To make this precise, we first review some standard concepts from parameterized complexity. The gold standard for tractability in parameterized complexity is membership in the class of fixed-parameter tractable (**FPT**) problems [25]. Roughly speaking, a factored problem will be fixed-parameter tractable (i.e. in **FPT**) if on an instance composed using k smaller instances of size at most n, there is an algorithm computing the function in $O(f(k)n^C)$ time for some fixed function f and absolute constant C independent of k. In these settings, we think of n as large and k as small, so that any dependence on k alone is preferable over an exponent of n that grows with k. A natural approach to solving problems on factored graphs is to explicitly compute the graph, requiring time and space $n^{O(k)}$, and solving the problem on the explicit graph. Such an algorithm has an exponent growing with k. Thus such factored problems will always be in **XP**[1], the class of problems that are polynomial for

[1] assuming the problem can be solved in polynomial time on the explicit graph

any particular k, but not necessarily in **FPT**. Even for simple problems with linear time algorithms (on explicit graphs), computing an explicit representation already requires $n^{O(k)}$ time and so this approach does not put the problem in **FPT**.

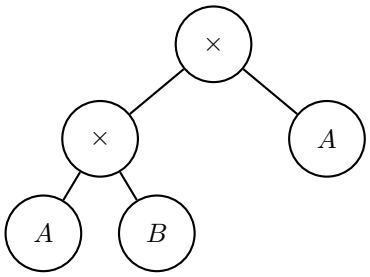
In this work, we initiate the study of the complexity of computational problems on factored graphs. In particular, we consider the question: for a given computational problem, is there an algorithm substantially better than first explicitly computing the factored graph G? This can be formalized as: is the problem in **FPT**, and, if not, how does the best exponent of n possible depend on k?

1.1 Definition of Factored Graphs

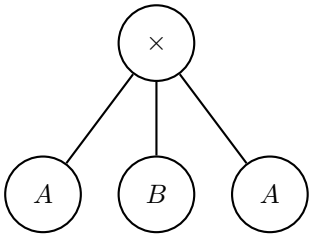
We consider a highly natural class of factored problems where the instances are graphs, and the operations are well-studied graph products and unions. Our factored graphs consist of arbitrary combinations of graphs under the following (binary) operations. Unless otherwise stated, all graphs are directed and (a, b) denotes an edge from a to b.

1. **Cartesian Product.** Given A, B, the Cartesian product $A \square B$ has vertices (a, b) for $a \in A, b \in B$ and there is an edge from (a, b) to (c, d) if either 1) $a = c$ and $(b, d) \in E(B)$, or 2) $(a, c) \in E(A)$ and $b = d$.
2. **Tensor Product.** Given A, B, the tensor product $A \times B$ has vertices (a, b) for $a \in A, b \in B$ and there is an edge from (a, b) to (c, d) if $(a, c) \in E(A)$ and $(b, d) \in E(B)$.
3. **Union.** Given A, B, the union $A \cup B$ has vertices $x \in V(A) \cup V(B)$ and there is an edge from x to y if $(x, y) \in E(A) \cup E(B)$.

A factored graph G can be defined by an ordered tree $T(G)$, where internal nodes are labelled by one of the above three operations $\square, \times, \cup$ and leaves are labelled with (possibly identical) input graphs. Since each operation is associative, we in fact allow the internal nodes to have arbitrary degree.[2] We say that a factored graph $G = f(G_1, \ldots, G_k)$ is *of complexity* (n, k) if the tree has k leaves and each input graph has at most n vertices. Here, we stress that the we parameterize by the number of leaves in $T(G)$, regardless of the fan-in of specific internal nodes or if a certain input graph appears in multiple leaf nodes. As an example, both $(A \times B) \times A$ and $A \times B \times A$ are factored graphs of complexity $(n, 3)$, where $n = \max(|V(A)|, |V(B)|)$. We illustrate their tree structures in Figure 1.



Tree structure of $(A \times B) \times A$.



Tree structure of $A \times B \times A$.

Figure 1 Tree structures of factored graphs $(A \times B) \times A$ and $A \times B \times A$. Both factored graphs have complexity $(n, 3)$ where $n = \max(|V(A)|, |V(B)|)$. Note that in this example the two factored graphs are in fact isomorphic.

[2] We require each internal node to have degree greater than 1, as otherwise the operation is the identity.

As another example, any graph G can be represented by a factored graph of complexity $(2, |E(G)|)$ by taking a union over all edges of the graph.

We define a *factored component* of G, denoted G_F, as follows. G_F is defined by an ordered tree $T(G_F)$, which is obtained from $T(G)$ by deleting every internal node v labeled by $\cup$ and attaching exactly one child of v to its parent. Another way to understand this is that, since the union and product operations obey the distributive law, we can recursively apply this law to convert a factored graph formula into a union of products. Then, each of these product terms is a factored component. The vertices of G_F are k_F-tuples (note $k_F \leq k$, where k_F is the number of leaves of $T(G_F)$ and k is the number of leaves of $T(G)$). We emphasize that the vertices are *flattened* tuples and do not preserve the topology of $T(G_F)$ beyond the order of the leaves. We call k_F the *dimension* of the factored component G_F. The edges of G_F are determined by $T(G_F)$ according to the definition of the graph operations $\square$ and $\times$.

The vertex and edge sets of G are then given by the union of the vertex and edge sets of the factored components. While a vertex may belong to multiple factored components, they must all have the same dimension, allowing us to define the dimension of a vertex (Definition 14). We also observe that G has at most 2^k factored components. We give a more formal definition of factored graphs as well as simple examples in Section 2.

1.2 Our Contributions

Our first result provides an *unconditional* hardness for a natural parameterized version of the well-studied Lexicographically First Maximal Independent Set (LFMIS) problem. The standard decision version of the LFMIS problem takes as input a graph $G = (V(G), E(G))$, where the vertices are indexed as $V(G) = \{0, 1, \ldots, |V(G)|-1\}$, and a special vertex $s \in V(G)$. The problem asks whether s belongs to the LFMIS of G. In the parameterized version, the input is given as a factored graph $G = f(G_1, \ldots, G_k)$ and the vertex indices are provided only for each graph factor $V(G_i) = \{0, 1, \ldots, |V(G_i)| - 1\}$. To recover the indices for $V(G)$, recall that each vertex in the factored graph G is a flattened tuple of numbers $(v_1, \ldots, v_{k'})$ for some $k' \leq k$, and therefore, we define the vertex indices to be given according to the standard lexicographic order of these tuples, with the index 0 given to the vertex with the lowest lexicographic order. In this work, we show that the LFMIS problem on factored graphs is **XP**-complete under **FPT**-reductions[3] and therefore *unconditionally* not in **FPT**, providing a lower bound on the best possible exponent as a function of k. LFMIS is a natural **P**-complete problem [18], studied by [58, 59, 60] including in the parallel setting [20, 50, 12, 8]. We use the **P**-completeness of this problem as an intuition for why the factored version might be difficult (**XP**-complete), but we know of no direct connection between **P**-completeness and hardness of the factored version.

▶ **Theorem 1** (**XP**-completeness of LFMIS). *The LFMIS problem on factored graphs is* **XP***-complete under* **FPT***-reductions and not fixed-parameter tractable. In particular, the LFMIS problem on a factored graph $G = f(G_1, \ldots, G_k)$ of complexity (n, k) requires $n^{\Omega(\sqrt{k})}$ time.*

While many works in parameterized complexity gave *conditional* lower bounds against problems in **FPT** [25], for example completeness for the **W**-hierarchy [25, 26, 28] or the **A**-hierarchy [32], *unconditional* lower bounds against natural problems in **FPT** are relatively rare [27]. Moreover, although it is known that **FPT** $\subsetneq$ **XP** via a diagonalization argument,

[3] See Definition 9.3 [27]. Also formally stated in Definition 20 in the context of factored graph problems.

the literature on **XP**-complete problems remains sparse. The "pebble game" problem was first introduced by Kasai, Adachi, and Iwata [45]. Its parameterized version, the "k-pebble game", was one of the earliest natural combinatorial problems shown to require $\Omega(n^k)$ time and be **XP**-complete [1]. As an application of the k-pebble game, some other game problems, such as the "cat and mouse game" [13] and the "k-peg game", have also been proven to be **XP**-complete via reductions from the k-pebble game problem [1]. More recently, Berkholz established an unconditional lower bound of $O(n^{(k-3)/12})$-time for the existential k-pebble game [6]. Our result provides yet another natural example of an **XP**-complete problem with an explicit lower bound of $n^{\Omega(\sqrt{k})}$.

In contrast to the LFMIS problem on factored graphs, we show that the well-studied problem of clique counting [51] is in **FPT**, solvable in a fixed polynomial of the input graph sizes (where the exponent possibly depends on the size of the clique of interest) times some function on the number of input graphs. Subgraph counting (in particular clique counting) has been studied by a long line of works [51, 16, 11, 42] in a variety of computational models [21, 24, 2, 56, 39]. On graphs with n vertices, s-cliques can be counted in $O(n^s)$ time (and $n^{\Omega(s)}$ time is necessary under standard hardness assumptions). We show that counting s-cliques on factored graphs of complexity (n, k) is in $O(g(s, k)n^s)$ time for some fixed (large) function g. Note that we do not show clique counting is in **FPT** with respect to the clique size parameter s, but only with respect to the complexity of the factored graph k.

▶ **Theorem 2** (Counting Clique Subgraphs is in **FPT**). *Let H be a clique on s vertices. Then, computing the number of exact copies of H in G, denoted $\#\mathbf{H}(G)$, is fixed-parameter tractable. In particular, there is an algorithm computing $\#\mathbf{H}(G)$ on factored graphs G of complexity (n, k) in $O\left(g(s, k)n^s\right)$ time for some function $g(s, k)$.*

Finally, we turn to the problem of reachability, one of the most fundamental computational problems on graphs. On an explicit graph, algorithms such as depth-first search (DFS) and breadth-first search (BFS) compute reachability in linear time. This raises the question: can reachability on factored graphs be computed efficiently? Moreover, reachability is closely related to classic space complexity classes. The general reachability problem is an important **NL**-complete problem [57, 43]; meanwhile, Cook and Mckenzie showed that reachability on a directed acyclic graph of outdegree at most one is **L**-complete [19]. This naturally gives rise to another question: does the completeness of reachability for a classic complexity class extend to its parameterized counterpart when the inputs are given as factored graphs, similar to what we have seen in the case of LFMIS in Theorem 1?

Indeed, we show that the parameterized version of reachability on factored graphs is **XNL**-complete under **FPT**-reductions, where **XNL** is the class of parameterized problems solvable in nondeterministic logarithmic space for any fixed parameter k. On the other hand, while we cannot provide a definitive answer to whether reachability on factored graphs is in **FPT**, we show that answering this question would exactly resolve a major open problem in classical complexity theory. Specifically, we establish a parameterized analog of the well-known open problem that asks whether $\mathbf{NL} \subseteq \mathbf{DTIME}(n^C)$ for some fixed constant C.

▶ **Theorem 3** (**XNL**-completeness of Reachability). *Reachability on factored graphs is* **XNL***-complete under* **FPT***-reductions. Furthermore, the following are equivalent:*

1. *There is a constant C such that $\mathbf{NL} \subseteq \mathbf{DTIME}(n^C)$.*
2. $\mathbf{XNL} \subseteq \mathbf{FPT}$.

This result also has several further implications. First, we observe that if reachability is not in **FPT**, $\mathbf{NSPACE}(h(n)) \not\subseteq \mathbf{P}$ for any space-constructible $h(n) = \omega(\log n)$. On the other hand, if reachability is in **FPT** then the Exponential Time Hypothesis (ETH) is false.

In particular, we claim that since k-SUM (determining if in a set of n numbers of $O(k \log n)$ bits there is a subset of k numbers summing to zero) is in **NL**, then k-SUM can be solved in $O\left(n^C\right) = n^{o(k)}$ time, and therefore ETH is false [55].[4] Finally, by combining the above implications, one can also conclude that if ETH is true, then $\mathbf{NSPACE}(\omega(\log n)) \subsetneq \mathbf{P}$, a result that, to the best of our knowledge, has not been previously established.

Parameterized space complexity classes, including **XL** and **XNL**, have also been studied extensively in the literature [10, 17, 33, 29]. Chen, Flum, and Grohe introduced the first complete problems for **XL** and **XNL** under parameterized-logspace-reductions[5] [17]. In a related direction, [9] identified a wide variety of problems complete for the class **XNLP**, which is the class of parameterized problems solvable in nondeterministic logarithmic space *and* polynomial time for each parameter. Most relevant to our result, Elberfeld, Stockhusen, and Tantau also showed that a parameterized version of reachability with multi-colored edges is complete for the class parameterized-**NL**-cert under **PL**-reductions [29].

1.3 Related Work

A long line of work, initiated by [25, 26, 28, 32], has studied the complexity of parameterized problems as a function of their input size and a parameter. Within parameterized complexity, a common theme is to study the complexity of problems given succinct representations of their input. For example, several previous works have investigated the complexity of computing Nash equilibrium of *succinct* games (represented implicitly) [31, 23, 22, 34, 53, 36]. As another example, [62] considers the complexity of combinatorial problems with succinct representation. Similarly, [52, 63, 40, 41, 49, 48, 14, 15] study the complexity of various computational problems on periodic structures, i.e., travel schedules on a periodic timetable.

Most relevant to this paper, several works have investigated computational problems on graphs with succinct representations such as small circuits [35], distributed graphs described by low complexity agents [4], and factored problems [21]. However, none of these works consider the complexity of factored graphs formed under graph products.

On the subject of succinct representations, researchers have also studied how to represent graphs as efficiently as possible [44, 38, 5, 30, 7, 47, 54].

2 Preliminaries

Unless otherwise noted, we work with directed graphs, with edges from a to b denoted by (a, b). For a graph G, we denote its vertices by $V(G)$ and edges by $E(G)$. For any subset of vertices $S \subset V(G)$, let $G[S]$ denote the subgraph induced by S. For a set X, we define $\mathcal{P}(X) = \{Y : Y \subseteq X\}$ to be the power set of X.

2.1 Graph Products

We begin with the definitions of the relevant graph operations. Let G and H be two directed graphs.

▶ **Definition 4.** *The* Cartesian product $G \square H$ *of* G *and* H *has vertex set* $V(G) \times V(H)$ *and directed edges* $((v_1, u_1), (v_2, u_2))$ *if and only if either*

- $v_1 = v_2$ *and* $(u_1, u_2) \in E(H)$*, or*
- $u_1 = u_2$ *and* $(v_1, v_2) \in E(G)$

[4] We need $O(k^2 \log n)$ bits to verify that k numbers of $O(k \log n)$ bits sum to zero.
[5] Also known as **PL**-reductions, a more restrictive notion compared to **FPT**-reductions

As a simple example, note that the Cartesian product of two paths is a grid.

▶ **Definition 5.** *The* tensor product $G \times H$ *of* G *and* H *has vertex set* $V(G) \times V(H)$ *and directed edges* $((v_1, u_1), (v_2, u_2))$ *if and only if* $(v_1, v_2) \in E(G)$ *and* $(u_1, u_2) \in E(H)$.

▶ **Definition 6.** *The* union $G \cup H$ *of* G *and* H *has vertex set* $V(G) \cup V(H)$ *and edge set* $E(G) \cup E(H)$.

While there are many other graph products and operations to consider, such as the or product, or graph negation, we will primarily focus our study of factored graphs under the above three operations.

We note that the above three operations are associative, and observe that the products of higher arity are given as follows.

▶ **Definition 7.** *The* Cartesian product $\square_{i=1}^{k} G_i$ *has vertex set* $\prod_{i=1}^{k} V(G_i)$ *and directed edges* $((v_1, \ldots, v_k), (u_1, \ldots, u_k))$ *if and only if there is some index* $j \in [k]$ *such that* $(v_j, u_j) \in E(G_j)$ *and* $v_i = u_i$ *for all* $i \neq j$.

▶ **Definition 8.** *The* tensor product $\times_{i=1}^{k} G_i$ *has vertex set* $\prod_{i=1}^{k} V(G_i)$ *and directed edges* $((v_1, \ldots, v_k), (u_1, \ldots, u_k))$ *if and only if* $(v_i, u_i) \in E(G_i)$ *for all* $i \in [k]$.

▶ **Definition 9.** *The* union $\bigcup_{i=1}^{k} G_i$ *of* G *and* H *has vertex set* $\bigcup_{i=1}^{k} V(G_i)$ *and edge set* $\bigcup_{i=1}^{k} E(G_i)$.

2.2 Factored Graph Construction

We now describe how the above operations construct factored graphs. Formally, we define a factored graph by describing how the above graph operations combine the input graphs into a single graph. This motivates the definition of a factored graph according to a tree which specifies the order of operations, or the tree structure of a factored graph.

▶ **Definition 10** (Factored Graph Tree Structure)**.** *The* factored graph tree structure *with* k *leaves, denoted* $f(G_1, \ldots, G_k)$*, is an ordered tree with* k *leaves where internal nodes are labelled by an operation (one of* $\square, \times$*, or* $\cup$*) and the leaves are labelled by graphs* $G_1, \ldots, G_k$*. Note that* G_i *are arbitrary and not necessarily distinct. We require that each internal node has degree at least* 2.

Let $G = f(G_1, \ldots, G_k)$ *denote that* G *is the* factored graph *given by the factored graph tree structure* $f(G_1, \ldots, G_k)$*. For convenience, let* $T(G)$ *denote the factored graph tree structure. We say that the factored graph tree structure* $f(G_1, \ldots, G_k)$*, or simply* G*, is* of complexity (n, k) *if* $T(G)$ *has at most* k *leaves and each leaf is labelled by a graph with at most* n *vertices.*

To illustrate our definitions, we will use the following example of a factored graph $G = ((A \times B) \cup C) \square (D \times (E \cup F))$. Note that there is a natural correspondence between the tree structures of factored graphs and the formulas of the above form, by using parentheses to delineate subtrees of the tree structure. The tree structure of G is given in Figure 2.

We now describe how the vertex and edge sets of a factored graph are obtained from its tree structure. First, we define factored components of factored graphs.

▶ **Definition 11** (Factored Component)**.** *Let* G *be a factored graph. A* factored component G_F *of* G *is the factored graph whose tree structure* $T(G_F)$ *is obtained from the tree structure of* G *by recursively replacing each internal node labelled by* $\cup$ *with the subtree rooted at one of its children. Note that the internal nodes of* $T(G_F)$ *are labelled either* $\square$ *or* $\times$.

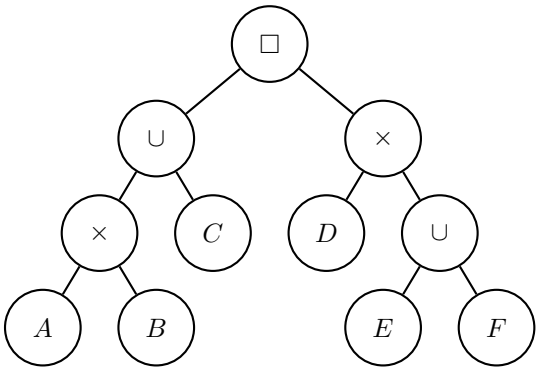
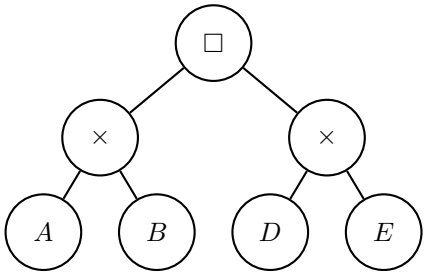


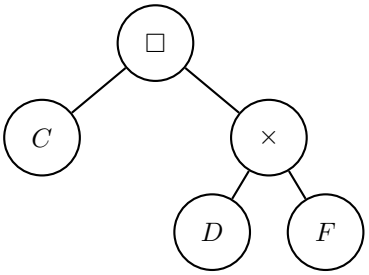
Figure 2 Tree structure of the factored graph $G = ((A \times B) \cup C) \,\square\, (D \times (E \cup F))$.

We say G_F has dimension ℓ *if $T(G_F)$ has ℓ leaves. Suppose the leaves are labelled $G_1, \ldots, G_\ell$. Then, $V(G_F) = \prod_{i=1}^{\ell} V(G_i)$. We say every vertex $v \in V(G_F)$ has dimension ℓ. The edge set $E(G_F)$ is determined by $T(G_F)$ and $E(G_1), \ldots, E(G_\ell)$ following the rules of Cartesian and tensor products.*

In our example, the factored components are $(A \times B) \,\square\, (D \times E), (A \times B) \,\square\, (D \times F), C \,\square\, (D \times E)$, and $C \,\square\, (D \times F)$, with dimensions $4, 4, 3, 3$, respectively. Some example tree structures of factored components are given in Figure 3.



Tree structure of $(A \times B) \,\square\, (D \times E)$.



Tree structure of $C \,\square\, (D \times F)$.

Figure 3 Examples of the Tree Structures of Factored Components.

Here, we emphasize that the vertex sets of the factored components are *flattened* products of the vertex sets of the graphs labeling the leaves. For example, the vertex of the factored component $(A \times B) \,\square\, (D \times E)$ has the form (a, b, d, e) not $((a, b), (d, e))$. In particular, the vertex sets do not depend on the topology of the tree beyond the order of the leaves. Finally, the vertex and edge set of G is simply the union of the factored components.

▶ **Definition 12** (Factored Graph). *Let $G = f(G_1, \ldots, G_k)$ be a factored graph with tree structure $T(G)$ and factored components $G_{F_1}, \ldots, G_{F_m}$. Define $V(G) = \bigcup_{i=1}^{m} V(G_{F_i})$ and $E(G) = \bigcup_{i=1}^{m} E(G_{F_i})$.*

We note here that different factored graphs can have identical vertex and edge sets. For example, $G_1 = (H_1 \times H_2) \times H_3$ and $G_2 = H_1 \times (H_2 \times H_3)$ both have vertex set $\prod_{i=1}^{3} H_i$ and edges from (h_1, h_2, h_3) to (h'_1, h'_2, h'_3) if and only if $(h_i, h'_i) \in E(H_i)$ for all i. This aligns with our expectation, as the tensor product is associative.

Note that a single vertex in G can belong to multiple factored components. In our running example, if there is a vertex $x \in V(E) \cap V(F)$, then the vertex (c, d, x) is in the factored components corresponding to $C \square (D \times E)$ and $C \square (D \times F)$. However, both factored components have dimension 3. This is formalized in the following lemma, which follows directly from the construction of factored graphs.

▶ **Lemma 13.** *Suppose $v \in V(G)$ is in factored components $G_F, G_{F'}$. Then,* $\dim(G_F) = \dim(G_{F'})$.

This allows us to define the dimension of a vertex $\dim(v)$ as the dimension of any factored component it belongs to.

▶ **Definition 14** (Vertex Dimension). *Let $G = f(G_1, \ldots, G_m)$ be a factored graph and $v \in V(G)$ be a vertex. The* dimension *of v, denoted by* $\dim(v)$*, is the dimension* $\dim(G_F)$ *of any factored component G_F that contains v.*

By construction, the endpoints of any edge must belong to the same factored component.

▶ **Lemma 15.** *Let $(u, v) \in E(G)$ be an edge. Then, u and v belong to the same factored component.*

As a corollary, we show that there are no edges between vertices of different dimensions, since vertices in the same factored component necessarily have the same dimension.

▶ **Corollary 16.** *Let (u, v) be an edge in G. Then, u and v have the same dimension.*

Finally, since every vertex has a unique dimension, it is clear that vertex dimension partitions the vertices of the factored graph. Thus, vertex dimension induces an equivalence relation on $V(G)$.

2.3 Parameterized Complexity

In this section, we review the parameterized complexity classes relevant to this paper with respect to problems on factored graphs. We begin with the definition of fixed-parameter tractability (**FPT**).

▶ **Definition 17** (Fixed-parameter Tractability (**FPT**) for Factored Graph Problems). *A problem on factored graphs is* fixed-parameter tractable *if there exists an algorithm that solves the problem on factored graph inputs of complexity (n, k) in time $O\left(g(k)n^C\right)$, where g is a function of k and C is a constant independent of k.*

We also study the parameterized complexity classes **XP** and **XNL**, which correspond to the parameterized version of the classical complexity classes **P** and **NL**, respectively. More generally, one can define the parameterized version **XC** for any classical complexity class **C** as follows [27]: a parameterized language $L \in \mathbf{XC}$ if and only if $L_k \in \mathbf{C}$ for every parameter k. Here, L_k is referred to as the *k-th slice* of L, which is the subset of L consisting of all instances with parameter k. We now give the definitions of **XP** and **XNL** in the context of factored graph problems.

▶ **Definition 18** (**XP** for Factored Graph Problems). *A problem on factored graphs is* in **XP** *if there exists an algorithm that solves the problem on factored graph inputs of complexity (n, k) in time $O\left(f(k)n^{f(k)}\right)$, where f is a function of k.*

▶ **Definition 19** (**XNL** for Factored Graph Problems). *A problem on factored graphs is* in **XNL** *if there exists a nondeterministic algorithm that solves the problem on factored graph inputs of complexity* (n, k) *in space* $O(f(k) \log n)$*, where* f *is a function of* k*.*

Finally, for **XP**- and **XNL**-completeness, we follow the usual definition that a problem is considered to be *complete* for a complexity class **C** if it belongs to **C** and is *hard* for **C** under some suitable notion of reductions. Here, we use the standard notion of parameterized mapping reduction (see Definition 9.3 in [27]), also known as the **FPT**-reduction [17], under which the class **FPT** is closed. We now give a definition for **XP**- and **XNL**-hardness under this reduction in the context of factored graph problems.

▶ **Definition 20** (**XP**-hardness (resp. **XNL**-hardness) for Factored Graph Problems). *A problem* L' *on factored graphs is* **XP**-hard *(resp.* **XNL**-hard*) if for every parameterized language* $L \in \mathbf{XP}$ *(resp.* $L \in \mathbf{XNL}$*), there exists a mapping* F *such that for every parameter* k*,* L_k *mapping reduces to* $L'_{k'}$ *under* F *using* $O(f(k)n^{O(1)})$ *time, where* f *is a function of* k *and* k' *depends only on* k*.*

3 Technical Overview

In this section, we provide a technical overview of our results. Due to space constraints, the proofs are left to the full version of the paper.

3.1 Lexicographically First Maximal Independent Set

Recall that our first main result is the **XP**-completeness of the LFMIS problem on factored graphs.

▶ **Theorem 1** (**XP**-completeness of LFMIS). *The LFMIS problem on factored graphs is* **XP***-complete under* **FPT***-reductions and not fixed-parameter tractable. In particular, the LFMIS problem on a factored graph* $G = f(G_1, \ldots, G_k)$ *of complexity* (n, k) *requires* $n^{\Omega(\sqrt{k})}$ *time.*

Theorem 1 is a consequence of a generic reduction from an arbitrary language $L \in \mathbf{DTIME}(n^\ell)$ to the LFMIS problem on factored graphs. Specifically, given a language $L \in \mathbf{DTIME}(n^\ell)$ and the corresponding Turing machine that decides L in $O(n^\ell)$ time, we construct a factored graph G of complexity $(O(n), O(\ell^2))$ using $O(\ell^2 n^2)$ time, such that solving the LFMIS on G simulates the Turing machine M. Then, it is fairly straightforward from this reduction that the problem is **XP**-complete. Moreover, we also obtain the explicit lower bound of $n^{\Omega(\sqrt{k})}$ by an application of the Time Hierarchy Theorem to the reduction.

Before proceeding to the detailed explanation, we first provide a high level intuition of this construction. Given a Turing machine, we can encapsulate its computation history in a matrix, where each row represents a configuration of the Turing machine at a specific time. The key idea of the reduction is to construct a graph in such a way that selecting vertices for the LFMIS corresponds to recovering the matrix entries, and thereby simulating the machine. To achieve this, we construct a graph with a grid structure mirroring the matrix. At each grid point, we place a collection of vertices representing all the possible choices for the corresponding matrix entry. The edges are defined according to the machine's transition function to ensure that the LFMIS chooses the single correct vertex from each grid point that agrees with the computation history of the Turing Machine, effectively allowing us to simulate the Turing machine by solving the LFMIS on the graph. It turns out that this graph has a highly regular structure and therefore can be factorized into a more succinct representation. In the remainder of this (sub)section, we explain this reduction in further detail.

Let L be a language in $\mathbf{DTIME}(n^\ell)$ and let M be the corresponding Turing machine that decides L in time $O(n^\ell)$. Given an input x to M, if M halts within time T, then the entire computation history of M on x can be represented by a $T \times T$ matrix W, where the i-th row of W represents the configuration of M at time i. To achieve this, each entry $W_{i,j}$ contains the following information about the j-th tape cell at time i: 1) the symbol occupying the cell, 2) whether the tape head is over the cell, and 3) the machine's current state if the tape head is over the cell. The goal now is to define a graph G such that the LFMIS of G recovers the computation history W, thereby simulating the machine M. We modify the Turing machine so that it suffices to query a single vertex in the graph for the halting state of M.

Explicit Graph Definition

We begin by defining G explicitly as an ordinary graph and then give a factored representation of G. Let S be the set of vertices corresponding to all possible choices for an entry in the matrix W. We define the vertex set of G to be T^2 copies of S, arranged in a $T \times T$ grid-like pattern analogous to the matrix W. Each copy of S is referred to as a *supernode*, and we use $S_{i,j}$ to denote the supernode in the i-th row and j-th column. Intuitively, the supernode $S_{i,j}$ represents all possible choices for the entry $W_{i,j}$. Therefore, we must define the edges of G such that the LFMIS includes the single correct vertex, which we call $w_{i,j}$, from each $S_{i,j}$ that agrees with the entry $W_{i,j}$.

The edge set of G consists of two types of edges: *intra*-supernode and *inter*-supernode edges. The intra-supernode edges are designed to ensure that the LFMIS contains *at most one* vertex within each supernode. This is easily achieved by defining a complete digraph on each supernode. On the other hand, the inter-supernode edges are constructed to ensure that the LFMIS contains *the correct* vertex from each supernode that agrees with the corresponding entry of W. This construction is more subtle and leverages the fact that the configuration at any given time of a deterministic machine uniquely determines the next configuration. Moreover, since the tape head can only move one step left or right at a time, each entry $W_{i,j}$ can be uniquely determined given only the three neighboring entries from the previous row: $W_{i-1,j-1}, W_{i-1,j}, W_{i-1,j+1}$ (instead of the entire previous row). To view this from another perspective: the entries $W_{i-1,j-1}$, $W_{i-1,j}$, and $W_{i-1,j+1}$ each restricts the possible choices for $W_{i,j}$ according to the machine's transition function. We define edges between neighboring supernodes to encode these restrictions. Specifically, for $v \in S_{i,j}$ and $v' \in S_{i+1,j'}$, a directed edge (v, v') indicates that, if v is chosen to represent the j-th tape cell at time i, then v' cannot represent the j'-th tape cell at the next time step $i + 1$, based on the machine's transition function. We hope that if the vertices $w_{i-1,j-1} \in S_{i-1,j-1}$, $w_{i-1,j} \in S_{i-1,j}$, and $w_{i-1,j+1} \in S_{i-1,j+1}$ are correctly contained in the LFMIS, then the edges ensure that $w_{i,j}$ is the unique vertex in $S_{i,j}$ that is not adjacent to $w_{i-1,j-1}, w_{i-1,j}$, or $w_{i-1,j+1}$. This will allow us to build the LFMIS inductively. The base case (which corresponds to the first row) is more technical and involves adding horizontal connections between supernodes.

Factored Graph Construction

Naively, this graph has $\Omega(T^2)$ vertices, where $T^2 = \Omega(n^{2\ell})$. However, since the computation rules of M are local (depending only on the symbol of the work tape under the head) and repetitive (the same rules apply regardless of the head's absolute position), we can give a succinct factored representation of G. Specifically, we show that G has a factored graph representation of complexity $(O(n), O(\ell^2))$. The factorization of G is outlined in Figure 4.

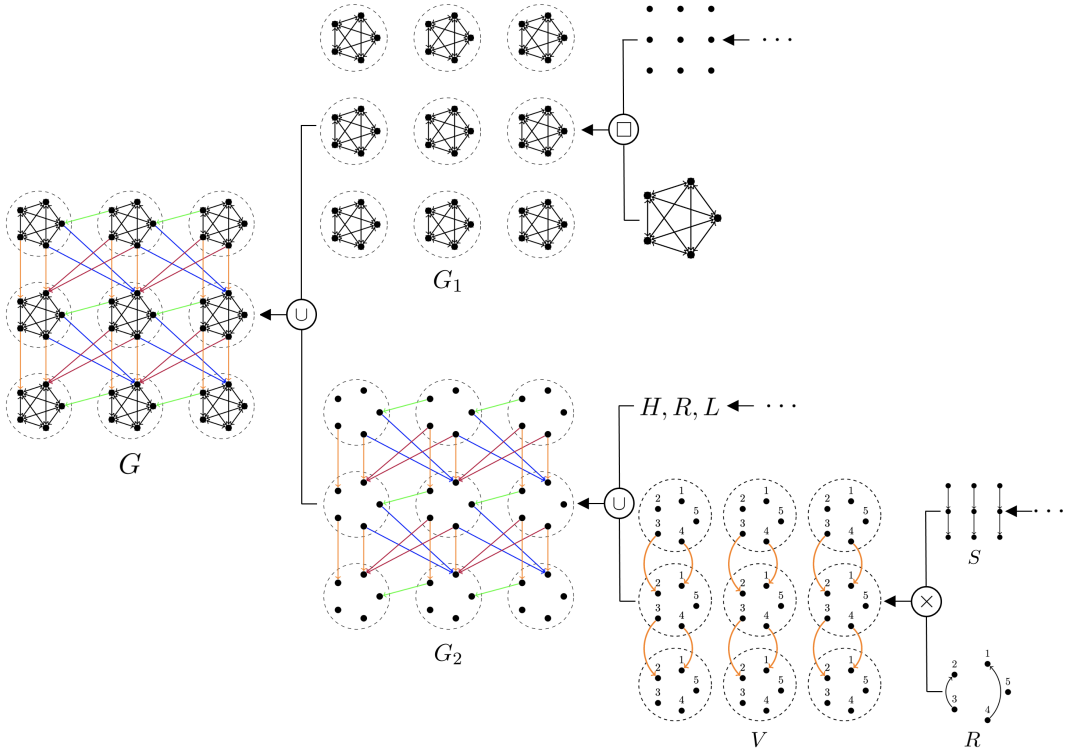


Figure 4 Overview of the Factorization of G. Supernodes are enclosed within dotted circles.

The key idea is to break down the graph G into regular and repetitive substructures. Following this, we begin by decomposing G into two subgraphs, G_1 and G_2, such that both subgraphs share the same vertex set as G, but G_1 contains only the intra-supernode edges and G_2 contains only the inter-supernode edges. G_1 forms a $T \times T$ grid of complete supernodes, which can be represented as the Cartesian product of an empty $T \times T$ grid of vertices and a single complete supernode. Similarly, G_2 is a $T \times T$ grid of supernodes, but with edges connecting neighboring supernodes in four possible directions: vertical (V), horizontal (H), diagonally-right R, and diagonally-left (L). We can further decompose G_2 into four subgraphs V, H, R, and L, which only contain the edges which are in the direction indicated by the name of the subgraph. Each of these subgraphs can be further decomposed into a "structure" graph and a "relation" graph using the tensor product. The structure graph S consists of a $T \times T$ grid of (ordinary) vertices with edges connecting all neighboring vertices in the corresponding direction of the subgraph. The relation graph R encodes the connections between vertices in neighboring supernodes in the corresponding direction. Moreover, the structure graphs themselves can be further factorized. For example, the structure graph S for the subgraph V can be expressed as the Cartesian product of a path of length T and an empty graph on T vertices. When the path length is a perfect power b^k (where $b, k \geq 1$ are integers), it can be further decomposed into a union of k factored graphs of complexities (b, k). Further details can be found in the full version of the paper.

3.2 Counting s-Cliques

In contrast to the first result, our second main result states that counting cliques on factored graphs is in **FPT**. For this result, we assume that all graphs are undirected.

▶ **Theorem 2** (Counting Clique Subgraphs is in **FPT**). *Let H be a clique on s vertices. Then, computing the number of exact copies of H in G, denoted $\#\mathbf{H}(G)$, is fixed-parameter tractable. In particular, there is an algorithm computing $\#\mathbf{H}(G)$ on factored graphs G of complexity (n,k) in $O\left(g(s,k)n^s\right)$ time for some function $g(s,k)$.*

In the technical overview, we use edge counting ($s = 2$) as an illustrative example. Note that if (u,v) is an edge, u, v must have the same dimension (Corollary 16), so we may count the edges in each dimension separately. Thus, fix a dimension $d \leq k$.

There are at most 2^k factored components of dimension d. While some edges may belong to multiple factored components, we can use the inclusion-exclusion principle to avoid double-counting such edges. In particular, it suffices to consider counting the number of edges in an arbitrary intersection of factored components $G_{F_1} \cap \ldots \cap G_{F_m}$. Note that the number of such intersections, while being a large function of k, is crucially independent of n.

Fix such an intersection and consider a pair of vertices u, v with $u = (u_1, \ldots, u_d)$ and $v = (v_1, \ldots, v_d)$. Note that by determining whether u_i, v_i are equal or adjacent for all $i \in [k]$, we can infer from the factored graph tree structure whether $(u,v) \in E(G)$. Thus, we can categorize all pairs of vertices into 2^{2k} groups based on the relations $u_i = v_i$ and $(u_i, v_i) \in E(G_i)$. Note also that these groups are disjoint. Next, we collect the subset of groups that form edges in the current intersection $G_{F_1} \cap \ldots \cap G_{F_m}$. To count the number of edges, it then suffices to sum up the sizes of each group in this collection. For a given group, we can determine its size in $O(kn^2)$ time since the relations on each coordinate are independent. In particular, we can count the number of pairs of vertices in each input graph satisfying the relevant constraints and take the product over all input graphs.

3.3 Reachability

Finally, we study the reachability problem on factored graphs.

▶ **Theorem 3** (**XNL**-completeness of Reachability). *Reachability on factored graphs is* **XNL***-complete under* **FPT***-reductions. Furthermore, the following are equivalent:*

1. *There is a constant C such that* $\mathbf{NL} \subseteq \mathbf{DTIME}(n^C)$.
2. $\mathbf{XNL} \subseteq \mathbf{FPT}$.

We begin with an overview of the proof for the **XNL**-completeness result, followed by that for the equivalence result. In fact, we will see that both proofs rely on the same major building blocks in slightly different ways.

XNL-completeness

We follow the standard framework for showing that reachability is **NL**-complete [57].

Membership in **XNL** can be established using the same algorithm that shows the ordinary reachability problem is in **NL**, even if the input is now given as a factored graph. As in the standard proof, we non-deterministically guess the next vertex in the path and thus only require $O(k \log n)$-space for factored graphs $G = f(G_1, \ldots, G_k)$ of complexity (n,k). Formally, each vertex in G can be specified by $O(k \log n)$ bits, since each vertex is a tuple of at most k coordinates and each coordinate is a vertex in an input graph G_i with at most n vertices. Thus, the algorithm requires $O(k \log n)$ bits to write down the current vertex and, since the factored graph has at most n^k vertices, at most $O(k \log n)$ bits to keep track of the number of steps taken so far.

For **XNL**-hardness, suppose we have some language $L \in \mathbf{XNL}$. Then, there exists a nondeterministic Turing machine M deciding L using $f(k) \log n$-space on inputs of length n and parameter k, for some function f of k. The standard reduction creates an explicit

configuration graph where each vertex encodes a tuple consisting of state, input and work tape positions, and a setting of the work tape. Since the work tape has length $f(k) \log n$, there are at least $n^{f(k)}$ distinct settings of the work tape. Thus, even though reachability is computable in linear time, the size of the graph already depends exponentially on $f(k)$. However, we are able to exploit the locality of Turing machine operations so that the configuration graph can in fact be encoded by a factored graph of complexity $(\text{poly}(n), \text{poly}(f(k)))$. Given some configuration of M on input x (a tuple of state, tape head locations and work tape contents), we split the configuration into segments, where each segment only contains $\log n$ contiguous bits of the work tape contents. Note that each segment, even if it encodes a state and tape head locations, only has $n^{O(1)}$ possible values, since one segment of the work tape only has $\log n$ bits. Thus, we can represent all possible configurations of a segment explicitly using a graph with $n^{O(1)}$ vertices, while a product of $f(k)$ segment graphs can explicitly represent any full configuration of M on input x.

It remains to express the appropriate transitions between configurations using a factored graph. There are two types of transitions: *intra*-segment transitions, where the work tape head stays within the same segment, and *inter*-segment transitions, where the work tape head moves from one segment to another (adjacent) segment. In either case, at most two segments of the work tape are active, where a segment is active during a transition if the work tape head either starts or ends in the segment and inactive otherwise. Thus, we can express individual segments (or pairs of segments) explicitly using input graphs of size $n^{O(1)}$. Since the active segments and inactive segments do not interact, we can encode all *intra*-segment transitions of a single segment using factored graphs of complexity $(n^{O(1)}, f(k))$. Similarly, we can encode all *inter*-segment transitions between a single pair of adjacent segments. Summing over all $O(f(k))$ segments and pairs of adjacent segments, we encode the configuration graph in a factored graph of complexity $(n^{O(1)}, O(f(k)^2))$.

We now briefly describe how the locality of Turing machine computation allows us to express the configuration graph of M on x using factored graphs. In the overview, we describe only *intra*-segment transitions. First, note that if the work tape head is not currently placed in a segment, the work tape contents of this segment cannot change. We thus define an *inactive* graph, where for every possible work tape content, we create a self-loop vertex. If the work tape head is currently placed in the segment, the work tape contents may change. Thus, we define an *active* graph with all possible configurations on a given segment as nodes, and edges encoding valid transitions. For a given segment, taking the tensor product of the active graph for this segment and the inactive graph for all other segments encodes all *intra*-segment transitions. A similar construction can be used to construct factored graphs that encode *inter*-segment transitions.

Equivalence to the Open Problem

For the forward direction, suppose there exists an absolute constant C such that $\mathbf{NL} \subseteq \mathbf{DTIME}(n^C)$. To show that $\mathbf{XNL} \subseteq \mathbf{FPT}$, it now suffices to show reachability on factored graphs is in $\mathbf{FPT}$ due to its $\mathbf{XNL}$-completeness under $\mathbf{FPT}$-reductions. Note that the algorithm used in the proof of $\mathbf{XNL}$-membership is in fact an $\mathbf{NL}$ algorithm for each fixed parameter k. Thus, if $\mathbf{NL} \subseteq \mathbf{DTIME}\left(n^C\right)$ for some C independent of k, then for any fixed k, reachability on factored graphs of complexity (n, k) can be computed in time $O(k^C n^{2C})$ (as factored graphs of this complexity have input size $O(kn^2)$) and is therefore in $\mathbf{FPT}$.

Conversely, if $\mathbf{XNL} \subseteq \mathbf{FPT}$, then in particular, reachability on factored graphs is in $\mathbf{FPT}$. Now, consider any language $L \in \mathbf{NL}$ with its associated nondeterministic Turing machine M that decides L in $S \log n$ space for some constant S. Using the same construction as in the

proof of **XNL**-hardness, we construct a factored graph G of complexity $(n^{O(1)}, O(S^2))$ where solving reachability on G simulates M. Since reachability on factored graphs is in **FPT**, it follows that reachability on G can be solved in time $O(f(S^2)n^{O(1)})$ for some function f, where the exponent on n is independent of S, hence independent of the specific language $L \in \mathbf{NL}$. This shows that $\mathbf{NL} \subseteq \mathbf{DTIME}(n^C)$ for some absolute constant C.

4 Conclusion and Future Work

We have studied the computational complexity of various problems on factored graphs. Even among problems with polynomial time algorithms on explicit graphs, we have shown that their parameterized complexity when the input is represented as a factored graph can differ quite drastically. In the context of parameterized complexity, we have sought to characterize which problems on factored graphs are in **FPT**. On the positive side, counting the number of copies of a small clique is in **FPT**. On the negative side, LFMIS is *unconditionally* not in **FPT**. Finally, we show that determining whether reachability is in **FPT** is equivalent to determining whether $\mathbf{NL} \subseteq \mathbf{DTIME}(n^{O(1)})$.

Can the unconditional lower bounds for LFMIS on factored graphs be used to prove similar unconditional lower bounds for other parameterized problems? One obstacle to doing this is the gap in complexity between the **P**-complete LFMIS problem and the easily parallelizable problems that form the bulk of the literature in parameterized complexity. However, reductions in fine-grained complexity often cut across traditional complexity classes, e.g., [46]. So we do not know a reason why this should not also be the case here. Either finding such unconditional results or explaining their impossibility would both be interesting. We could also hope to prove similar results for other **P**-complete problems.

While our lower bounds separate the problems of interest from **FPT**, they do not rule out significant improvements on the naive $n^{O(k)}$ algorithm of computing the factored graph G explicitly and solving the problem on G itself. For example, our LFMIS lower bound only unconditionally rules out algorithms with time $n^{o(\sqrt{k})}$. An interesting open problem is to provide a more *fine-grained* analysis of the complexity of factored graph problems, possibly distinguishing between the number of product and union operations.

In this work, we have chosen to study factored graphs under graph products and unions, specifically the Cartesian and tensor products. A natural extension is to consider other products and operations on graphs, or other interesting objects. Factored problems on bit strings were introduced in [21], and implicitly on integer-valued vectors in [3]. Do similar results hold for factored problems for these input domains and operations?

References

1 Akeo Adachi, Shigeki Iwata, and Takumi Kasai. Some combinatorial game problems require omega(n^k) time. *J. ACM*, 31(2):361–376, March 1984. doi:10.1145/62.322433.

2 Enric Boix Adserà, Matthew S. Brennan, and Guy Bresler. The average-case complexity of counting cliques in erdős-rényi hypergraphs. *CoRR*, abs/1903.08247:FOCS19–39, 2019. doi:10.48550/arXiv.1903.08247.

3 Haozhe An, Mohit Gurumukhani, Russell Impagliazzo, Michael Jaber, Marvin Künnemann, and Maria Paula Parga Nina. The fine-grained complexity of multi-dimensional ordering properties. *Algorithmica*, 84(11):3156–3191, 2022. doi:10.1007/s00453-022-01014-x.

4 Sanjeev Arora, David Steurer, and Avi Wigderson. Towards a study of low-complexity graphs. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikoletseas, and Wolfgang Thomas, editors, *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I*, volume 5555 of *Lecture Notes in Computer Science*, pages 119–131. Springer, Springer, 2009. doi:10.1007/978-3-642-02927-1_12.

5 Jérémy Barbay, Luca Castelli Aleardi, Meng He, and J. Ian Munro. Succinct representation of labeled graphs. In Takeshi Tokuyama, editor, *Algorithms and Computation, 18th International Symposium, ISAAC 2007, Sendai, Japan, December 17-19, 2007, Proceedings*, volume 4835 of *Lecture Notes in Computer Science*, pages 316–328. Springer, Springer, 2007. doi:10.1007/978-3-540-77120-3_29.

6 Christoph Berkholz. Lower bounds for existential pebble games and k-consistency tests. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25-28, 2012*, pages 25–34. IEEE Computer Society, 2012. doi:10.1109/LICS.2012.14.

7 Guy E. Blelloch and Arash Farzan. Succinct representations of separable graphs. In Amihood Amir and Laxmi Parida, editors, *Combinatorial Pattern Matching, 21st Annual Symposium, CPM 2010, New York, NY, USA, June 21-23, 2010. Proceedings*, volume 6129 of *Lecture Notes in Computer Science*, pages 138–150. Springer, Springer, 2010. doi:10.1007/978-3-642-13509-5_13.

8 Guy E. Blelloch, Jeremy T. Fineman, and Julian Shun. Greedy sequential maximal independent set and matching are parallel on average. In Guy E. Blelloch and Maurice Herlihy, editors, *24th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA '12, Pittsburgh, PA, USA, June 25-27, 2012*, pages 308–317. ACM, 2012. doi:10.1145/2312005.2312058.

9 Hans L. Bodlaender, Carla Groenland, Jesper Nederlof, and Céline M. F. Swennenhuis. Parameterized problems complete for nondeterministic FPT time and logarithmic space. *CoRR*, abs/2105.14882, 2021. doi:10.48550/arXiv.2105.14882.

10 Liming Cai, Jianer Chen, Rodney G. Downey, and Michael R. Fellows. Advice classes of parameterized tractability. *Ann. Pure Appl. Log.*, 84(1):119–138, 1997. Asian Logic Conference. doi:10.1016/S0168-0072(95)00020-8.

11 Chris Calabro, Russell Impagliazzo, Valentine Kabanets, and Ramamohan Paturi. The complexity of unique k-sat: An isolation lemma for k-cnfs. *J. Comput. Syst. Sci.*, 74(3):386–393, 2008. doi:10.1016/j.jcss.2007.06.015.

12 Neil J. Calkin, Alan M. Frieze, and Ludek Kucera. On the expected performance of a parallel algorithm for finding maximal independent subsets of a random graph. *Random Struct. Algorithms*, 3(2):215–222, 1992. doi:10.1002/rsa.3240030210.

13 Ashok K. Chandra and Larry J. Stockmeyer. Alternation. In *17th Annual Symposium on Foundations of Computer Science, Houston, Texas, USA, 25-27 October 1976*, pages 98–108. IEEE Computer Society, 1976. doi:10.1109/SFCS.1976.4.

14 Hubie Chen. Periodic constraint satisfaction problems: Polynomial-time algorithms. In Francesca Rossi, editor, *Principles and Practice of Constraint Programming - CP 2003, 9th International Conference, CP 2003, Kinsale, Ireland, September 29 - October 3, 2003, Proceedings*, volume 2833 of *Lecture Notes in Computer Science*, pages 199–213. Springer, Springer, 2003. doi:10.1007/978-3-540-45193-8_14.

15 Hubie Chen. Periodic constraint satisfaction problems: Tractable subclasses. *Constraints An Int. J.*, 10(2):97–113, 2005. doi:10.1007/s10601-005-0551-z.

16 Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.*, 72(8):1346–1367, 2006. doi:10.1016/j.jcss.2006.04.007.

17 Yijia Chen, Jörg Flum, and Martin Grohe. Bounded nondeterminism and alternation in parameterized complexity theory. In *18th Annual IEEE Conference on Computational Complexity (Complexity 2003), 7-10 July 2003, Aarhus, Denmark*, pages 13–29. IEEE Computer Society, 2003. doi:10.1109/CCC.2003.1214407.

18 Stephen A. Cook. A taxonomy of problems with fast parallel algorithms. *Inf. Control.*, 64(1-3):2–21, 1985. doi:10.1016/S0019-9958(85)80041-3.

19 Stephen A. Cook and Pierre McKenzie. Problems complete for deterministic logarithmic space. *J. Algorithms*, 8(3):385–394, 1987. doi:10.1016/0196-6774(87)90018-6.

20 Don Coppersmith, Prabhakar Raghavan, and Martin Tompa. Parallel graph algorithms that are efficient on average. In *28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987*, pages 260–269. IEEE, IEEE Computer Society, 1987. doi:10.1109/SFCS.1987.46.

21 Mina Dalirrooyfard, Andrea Lincoln, and Virginia Vassilevska Williams. New techniques for proving fine-grained average-case hardness. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 774–785. IEEE, IEEE, 2020. doi:10.1109/FOCS46700.2020.00077.

22 Constantinos Daskalakis, Alex Fabrikant, and Christos H. Papadimitriou. The game world is flat: The complexity of nash equilibria in succinct games. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part I*, volume 4051 of *Lecture Notes in Computer Science*, pages 513–524. Springer, Springer, 2006. doi:10.1007/11786986_45.

23 Konstantinos Daskalakis and Christos H. Papadimitriou. The complexity of games on highly regular graphs. In Gerth Stølting Brodal and Stefano Leonardi, editors, *Algorithms - ESA 2005, 13th Annual European Symposium, Palma de Mallorca, Spain, October 3-6, 2005, Proceedings*, volume 3669 of *Lecture Notes in Computer Science*, pages 71–82. Springer, Springer, 2005. doi:10.1007/11561071_9.

24 Laxman Dhulipala, Quanquan C. Liu, Julian Shun, and Shangdi Yu. Parallel batch-dynamic k-clique counting. In Michael Schapira, editor, *2nd Symposium on Algorithmic Principles of Computer Systems, APOCS 2020, Virtual Conference, January 13, 2021*, pages 129–143. SIAM, SIAM, 2021. doi:10.1137/1.9781611976489.10.

25 Rodney G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness I: basic results. *SIAM J. Comput.*, 24(4):873–921, 1995. doi:10.1137/S0097539792228228.

26 Rodney G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness II: on completeness for W[1]. *Theor. Comput. Sci.*, 141(1&2):109–131, 1995. doi:10.1016/0304-3975(94)00097-3.

27 Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999. doi:10.1007/978-1-4612-0515-9.

28 Rodney G. Downey, Michael R. Fellows, and Kenneth W. Regan. Descriptive complexity and the W hierarchy. In Paul Beame and Samuel R. Buss, editors, *Proof Complexity and Feasible Arithmetics, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, April 21-24, 1996*, volume 39 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 119–134. Citeseer, DIMACS/AMS, 1996. doi:10.1090/dimacs/039/07.

29 Michael Elberfeld, Christoph Stockhusen, and Till Tantau. On the space complexity of parameterized problems. In Dimitrios M. Thilikos and Gerhard J. Woeginger, editors, *Parameterized and Exact Computation - 7th International Symposium, IPEC 2012, Ljubljana, Slovenia, September 12-14, 2012. Proceedings*, volume 7535 of *Lecture Notes in Computer Science*, pages 206–217, Berlin, Heidelberg, 2012. Springer. doi:10.1007/978-3-642-33293-7_20.

30 Arash Farzan and J. Ian Munro. Succinct representations of arbitrary graphs. In Dan Halperin and Kurt Mehlhorn, editors, *Algorithms - ESA 2008, 16th Annual European Symposium, Karlsruhe, Germany, September 15-17, 2008. Proceedings*, volume 5193 of *Lecture Notes in Computer Science*, pages 393–404. Springer, Springer, 2008. doi:10.1007/978-3-540-87744-8_33.

31 Joan Feigenbaum, Daphne Koller, and Peter W. Shor. A game-theoretic classification of interactive complexity classes. In *Proceedings of the Tenth Annual Structure in Complexity Theory Conference, Minneapolis, Minnesota, USA, June 19-22, 1995*, pages 227–237. IEEE, IEEE Computer Society, 1995. doi:10.1109/SCT.1995.514861.

32 Jörg Flum and Martin Grohe. Fixed-parameter tractability, definability, and model-checking. *SIAM J. Comput.*, 31(1):113–145, 2001. doi:10.1137/S0097539799360768.

33 Jörg Flum and Martin Grohe. Describing parameterized complexity classes. *Inf. Comput.*, 187(2):291–319, 2003. doi:10.1016/S0890-5401(03)00161-5.

34 Lance Fortnow, Russell Impagliazzo, Valentine Kabanets, and Christopher Umans. On the complexity of succinct zero-sum games. *Comput. Complex.*, 17(3):353–376, 2008. doi:10.1007/s00037-008-0252-2.

35 Hana Galperin and Avi Wigderson. Succinct representations of graphs. *Inf. Control.*, 56(3):183–198, 1983. doi:10.1016/S0019-9958(83)80004-7.

36 Gianluigi Greco, Enrico Malizia, Luigi Palopoli, and Francesco Scarcello. The complexity of the nucleolus in compact games. *ACM Trans. Comput. Theory*, 7(1):3:1–3:52, 2014. doi:10.1145/2692372.2692374.

37 Shreya Gupta, Boyang Huang, Russell Impagliazzo, Stanley Woo, and Christopher Ye. The computational complexity of factored graphs. *CoRR*, abs/2407.19102, 2024. doi:10.48550/arXiv.2407.19102.

38 Xin He, Ming-Yang Kao, and Hsueh-I Lu. Linear-time succinct encodings of planar graphs via canonical orderings. *SIAM J. Discret. Math.*, 12(3):317–325, 1999. doi:10.1137/S0895480197325031.

39 Monika Henzinger, Andrea Lincoln, and Barna Saha. The complexity of average-case dynamic subgraph counting. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 459–498. SIAM, SIAM, 2022. doi:10.1137/1.9781611977073.23.

40 Franz Höfting and Egon Wanke. Polynomial algorithms for minimum cost paths in periodic graphs. In Vijaya Ramachandran, editor, *Proceedings of the Fourth Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, 25-27 January 1993, Austin, Texas, USA*, pages 493–499. ACM/SIAM, 1993. URL: http://dl.acm.org/citation.cfm?id=313559.313868.

41 Bruce Hoppe and Éva Tardos. The quickest transshipment problem. *Math. Oper. Res.*, 25(1):36–62, 2000. doi:10.1287/moor.25.1.36.15211.

42 Shweta Jain and C. Seshadhri. The power of pivoting for exact clique counting. In James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang, editors, *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 268–276. ACM, 2020. doi:10.1145/3336191.3371839.

43 Neil D. Jones. Space-bounded reducibility among combinatorial problems. *J. Comput. Syst. Sci.*, 11(1):68–85, 1975. doi:10.1016/S0022-0000(75)80050-X.

44 Sampath Kannan, Moni Naor, and Steven Rudich. Implicit representation of graphs. In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 334–343. ACM, 1988. doi:10.1145/62212.62244.

45 Takumi Kasai, Akeo Adachi, and Shigeki Iwata. Classes of pebble games and complete problems. *SIAM J. Comput.*, 8(4):574–586, 1979. doi:10.1137/0208046.

46 Marvin Künnemann, Ramamohan Paturi, and Stefan Schneider. On the fine-grained complexity of one-dimensional dynamic programming. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 21:1–21:15. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.21.

47 Sebastian Maneth and Fabian Peternek. A survey on methods and systems for graph compression. *CoRR*, abs/1504.00616, 2015. arXiv:1504.00616, doi:10.48550/arXiv.1504.00616.

48 Madhav V. Marathe, Harry B. Hunt III, Daniel J. Rosenkrantz, and Richard Edwin Stearns. Theory of periodically specified problems: Complexity and approximability. In *Proceedings of the 13th Annual IEEE Conference on Computational Complexity, Buffalo, New York, USA, June 15-18, 1998*, page 106. IEEE, IEEE Computer Society, 1998. doi:10.1109/CCC.1998.694596.

49 Madhav V. Marathe, Harry B. Hunt III, Richard Edwin Stearns, and Venkatesh Radhakrishnan. Approximation schemes for pspace-complete problems for succinct specifications (preliminary version). In Frank Thomson Leighton and Michael T. Goodrich, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 468–477. ACM, 1994. doi:10.1145/195058.195233.

50 Satoru Miyano. The lexicographically first maximum subgraph problems: P-completeness and NC algorithms. *Math. Syst. Theory*, 22(1):47–73, 1989. doi:10.1007/BF02088292.

51 Jaroslav Nešetřil and Svatopluk Poljak. On the complexity of the subgraph problem. *Commentationes Mathematicae Universitatis Carolinae*, 026(2):415–419, 1985. URL: http://eudml.org/doc/17394.

52 James B. Orlin. The complexity of dynamic languages and dynamic optimization problems. In *Proceedings of the 13th Annual ACM Symposium on Theory of Computing, May 11-13, 1981, Milwaukee, Wisconsin, USA*, pages 218–227. ACM, 1981. doi:10.1145/800076.802475.

53 Christos H. Papadimitriou and Tim Roughgarden. Computing correlated equilibria in multi-player games. *J. ACM*, 55(3):14:1–14:29, 2008. doi:10.1145/1379759.1379762.

54 Victor Parque and Tomoyuki Miyashita. On succinct representation of directed graphs. In *2017 IEEE International Conference on Big Data and Smart Computing, BigComp 2017, Jeju Island, South Korea, February 13-16, 2017*, pages 199–205. IEEE, IEEE, 2017. doi:10.1109/BIGCOMP.2017.7881738.

55 Mihai Patrascu and Ryan Williams. On the possibility of faster SAT algorithms. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1065–1075. SIAM, SIAM, 2010. doi:10.1137/1.9781611973075.86.

56 Jessica Shi, Laxman Dhulipala, and Julian Shun. Parallel clique counting and peeling algorithms. In Michael Bender, John Gilbert, Bruce Hendrickson, and Blair D. Sullivan, editors, *Proceedings of the 2021 SIAM Conference on Applied and Computational Discrete Algorithms, ACDA 2021, Virtual Conference, July 19-21, 2021*, pages 135–146. SIAM, SIAM, 2021. doi:10.1137/1.9781611976830.13.

57 Michael Sipser. Introduction to the theory of computation. *SIGACT News*, 27(1):27–29, 1996. doi:10.1145/230514.571645.

58 Ryuhei Uehara. A measure of parallelization for the lexicographically first maximal subgraph problems. In Rolf H. Möhring, editor, *Graph-Theoretic Concepts in Computer Science, 23rd International Workshop, WG '97, Berlin, Germany, June 18-20, 1997, Proceedings*, volume 1335 of *Lecture Notes in Computer Science*, pages 333–341. Springer, Springer, 1997. doi:10.1007/BFb0024508.

59 Ryuhei Uehara. Another measure for the lexicographically first maximal subgraph problems and its threshold value on a random graph. In *1999 International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '99), 23-25 June 1999, Fremantle, Australia*, pages 350–355. IEEE, IEEE Computer Society, 1999. doi:10.1109/ISPAN.1999.778963.

60 Ryuhei Uehara. A measure for the lexicographically first maximal independent set problem and its limits. *Int. J. Found. Comput. Sci.*, 10(4):473–482, 1999. doi:10.1142/S0129054199000332.

61 Emanuele Viola and Avi Wigderson. Local expanders. *Comput. Complex.*, 27(2):225–244, 2018. doi:10.1007/s00037-017-0155-1.

62 Klaus W. Wagner. The complexity of combinatorial problems with succinct input representation. *Acta Informatica*, 23(3):325–356, 1986. doi:10.1007/BF00289117.

63 Egon Wanke. Paths and cycles in finite periodic graphs. In Andrzej M. Borzyszkowski and Stefan Sokolowski, editors, *Mathematical Foundations of Computer Science 1993, 18th International Symposium, MFCS'93, Gdansk, Poland, August 30 - September 3, 1993, Proceedings*, volume 711 of *Lecture Notes in Computer Science*, pages 751–760. Springer, Springer, 1993. doi:10.1007/3-540-57182-5_66.

Error Correction for Message Streams

Meghal Gupta
University of California Berkeley, CA, USA

Rachel Yun Zhang
MIT, Cambridge, MA, USA

Abstract

In the setting of error correcting codes, Alice wants to send a message $x \in \{0,1\}^n$ to Bob via an encoding $\mathsf{enc}(x)$ that is resilient to error. In this work, we investigate the scenario where Bob is a *low space* decoder. More precisely, he receives Alice's encoding $\mathsf{enc}(x)$ bit-by-bit and desires to compute some function $f(x)$ in low space. A generic error-correcting code does not accomplish this because decoding is a very global process and requires at least linear space. Locally decodable codes partially solve this problem as they allow Bob to learn a given bit of x in low space, but not compute a generic function f.

Our main result is an encoding and decoding procedure where Bob is still able to compute any such function f in low space when a constant fraction of the stream is corrupted. More precisely, we describe an encoding function $\mathsf{enc}(x)$ of length $\mathrm{poly}(n)$ so that for any decoder (streaming algorithm) A that on input x computes $f(x)$ in space s, there is an explicit decoder B that computes $f(x)$ in space $s \cdot \mathrm{polylog}(n)$ as long as there were not more than $\frac{1}{4} - \varepsilon$ fraction of (adversarial) errors in the input stream $\mathsf{enc}(x)$.

2012 ACM Subject Classification Mathematics of computing → Coding theory

Keywords and phrases error-correcting codes, streaming algorithms, space-efficient algorithms

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.59

Funding *Meghal Gupta*: Supported by an NSF GRFP Fellowship.
Rachel Yun Zhang: Supported in part by DARPA under Agreement No. HR00112020023, an NSF grant CNS-2154149, and NSF Graduate Research Fellowship 2141064.

1 Introduction

Consider the following scenario: Alice streams a message denoted $x = x_1 \dots x_n$ to Bob that he receives and processes bit-by-bit. His goal is to compute some function $f(x_1 \dots x_n)$ unknown to Alice in significantly less space than necessary to store the entire stream. This scenario arises for instance in automatic control applications, where a device receives an incoming stream of data and needs to use it to make decisions.

As an example, a class of functions that Bob may wish to compute in small space from a message stream is the class of *linear functions*. If Bob's function is $f(x) = f_y(x) = x \cdot y \mod 2$ for some $y \in \{0,1\}^n$, then after receiving each bit of x, Bob adds $x_i \cdot y_i$ to a running sum. Bob only needs to track i and the running sum modulo 2, which is in total $\log n + 1$ bits of space.

However, this and other small space algorithms are very rigid when it comes to errors in the stream. Corruption of even one bit of Alice's message can change the output of Bob's linear function. The same applies if Bob's function is an index function, a majority, or the result of a decision tree.

In this work, we study what happens when there is noise in the incoming stream. In particular, we ask if it's possible to convert a given algorithm that processes a message stream into one that is robust to errors in the message while still allowing the output to be computed in low space.


© Meghal Gupta and Rachel Yun Zhang;
licensed under Creative Commons License CC-BY 4.0



In the usual error correction setting, Alice could just encode her message using a generic large distance error-correcting code. Bob would be able to compute any function f in the presence of a small constant fraction of error, but he must receive and store the whole stream in order to decode, which is far too much space. Even if Alice sends a locally decodable code, which has the property that to determine a single bit of x one only needs to track a few random bits of the codeword, it is not clear how Bob can compute a function requiring all n bits without $\Omega(n)$ storage. Our question is whether there is an encoding that preserves the space complexity of Bob's original decoding algorithm while being resilient to error.

In this work, we answer the question in the affirmative. Our main result is a scheme that protects any streaming algorithm against noise in the incoming stream. More precisely, we give an encoding $\mathsf{enc}(x)$ of length $O(n^{4+\delta})$ such that any streaming algorithm running in s space on the noiseless stream can be converted to one running in $s \cdot \text{polylog}(n)$ space on the encoded stream. It is correct with high probability whenever at most $\frac{1}{4} - \varepsilon$ of the stream was corrupted. In the specific case where Bob's function is a linear function such as dot product, our encoding can be made to have length $O(n^{2+\delta})$.

1.1 The Model

We provide a formal definition of a noise resilient transformation for a message stream and processing algorithm. The transformation has two components:

- An encoding function $\mathsf{enc}(\cdot) : X \subseteq \{0,1\}^n \to \{0,1\}^m$.
- An explicit transformation B that takes as input a deterministic streaming algorithm $A : X \to \mathbb{F}_q$ and outputs a randomized streaming algorithm $B_A = B(A) : \{0,1\}^m \to \mathbb{F}_q$.

The algorithm is said to be α-error resilient if whenever $\Delta(z, \mathsf{enc}(x)) < \alpha \cdot m$, then $B_A(z)$ outputs $A(x)$ with high probability.

1.2 Our Results

Our main result is a noise-resilient conversion for deterministic streaming algorithms.

▶ **Theorem 1.** *Fix $\varepsilon, \delta > 0$. For any $X \subseteq \{0,1\}^n$, there is a noise resilient transformation consisting of an encoding $\mathsf{enc} : X \subseteq \{0,1\}^n \to \{0,1\}^m$, and a transformation of algorithms B that is $\left(\frac{1}{4} - \varepsilon\right) \cdot m$ error resilient with probability $\geq 1 - 2^{O_\varepsilon(\log^2 n)}$. Moreover, $m = O_\varepsilon(n^{4+\delta})$, and for any deterministic algorithm A on domain X that runs in space s and time t, the algorithm $B(A)$ runs in space $s \cdot O_\varepsilon\left((\log n)^{O(1/\delta)}\right)$ and time $m \cdot O_{\varepsilon,\delta}\left(1 + \frac{t}{n^2}\right)$.*

In other words, given an algorithm A that accepts a stream of length n and uses s space, we demonstrate a noise resilient algorithm $B(A)$ computing A that uses $s \cdot \text{polylog}(n)$ space. The transformation enc of the stream x from the sender is independent of A and has length $O(n^{4+\delta})$.

A priori, it is not obvious that there is *any* low-memory noise resilient transformation of the algorithm A even with an arbitrary blow-up in communication. For example, if the sender were to encode their message using an arbitrary error-correcting code, the receiver would need to store the entire message in memory in order to decode it before applying the algorithm A. Our result shows that not only does this low-memory transformation exist, but that it can be done efficiently (both with respect to communication complexity and computational complexity).

Linear Streaming Algorithms

In the above transformation, the length of the resulting stream is $O(n^{4+\delta})$ for any $\delta > 0$. In the specific case where the streaming algorithm is *linear*, we construct a scheme where the stream length is only quadratic, namely $O(n^{2+\delta})$. The key property of linear streaming algorithms that we will be leveraging is that they can be computed in pieces in any order, and later computations do not depend on the results of previous ones.

▶ **Definition 2** (Linear Streaming Algorithms). *A linear streaming algorithm $A : \{0,1\}^n \to \mathbb{F}_q$ is described by a list of functions $g_i : \{0,1\} \to \mathbb{F}_q$ for $i \in [n]$, and computes the value $A(x) = g_1(x_1) + \ldots + g_n(x_n)$ (where addition is over $\mathbb{F}_q$) by tracking the partial sum $g_1(x_1) + \ldots + g_i(x_i)$ upon receiving the i'th bit.*

Note, for example, that every linear function on codomain $\{0,1\}^n$ admits a linear streaming algorithm. We note that linear streaming algorithms capture a large class of interesting functions, including linear sketching algorithms (see for example [12, 10, 1, 37]). For linear streaming algorithms, we show the following result.

▶ **Theorem 3.** *Fix $\varepsilon, \delta > 0$. There is a function $\mathsf{enc} : X \subseteq \{0,1\}^n \to \{0,1\}^m$ where $m = O_\varepsilon(n^{2+\delta})$ and an explicit transformation B such that the following holds: For any linear streaming algorithm A that takes as input $x \in X \subseteq \{0,1\}^n$ as a stream, runs in space s and time t, and outputs $A(x)$, the algorithm $B_A = B(A)$ takes as input $z \in \{0,1\}^m$ as a stream, runs in space $s \cdot O_\varepsilon((\log n)^{O(1/\delta)})$ and time $m \cdot O_{\varepsilon,\delta}\left(1 + \frac{t}{n^2}\right)$, and satisfies that whenever $\Delta(z, \mathsf{enc}(x)) < \left(\frac{1}{4} - \varepsilon\right) \cdot m$ then $B_A(z)$ outputs $A(x)$ with probability $\geq 1 - 2^{O_\varepsilon(\log^2 n)}$.*

Randomized Algorithms

We remark that our transformations in Theorems 1 and 3 are only for deterministic algorithms. However, this easily implies the result for algorithms A that *pre-sample* all of their randomness (that is, algorithms that fix their randomness before receiving any bits of x, at which point they are deterministic for the remainder of the algorithm). Such algorithms can be viewed as sampling from a distribution over deterministic algorithms. Our transformation can then be applied to the ensuing deterministic algorithm, thus correctly computing the function A with high probability while being resilient to noise.

We remark that this notion of randomized algorithms whose only access to randomness is pre-sampled at the start of computation is quite general: randomized algorithms that sample its randomness online can often be converted to protocols where the randomness is pre-sampled, see e.g. [44].

Non-Binary Input Alphabets

Our main theorems are stated for the setting where the input stream is binary. We remark that this assumption is without loss of generality: a larger alphabet stream can be converted to a binary one by assigning each alphabet symbol to a unique binary string.

1.3 Discussion of Subsequent Work

Since this paper was first posted, the work of [29] improved upon our schemes in a few major ways.

- For the case of general deterministic streaming algorithms, they demonstrate a transformation that only requires near-quadratic encoding length.

- They show that this is essentially tight: there is no transformation that has sub-quadratic encoding length.
- And, in the case of linear streaming algorithms, they construct a transformation that has an encoding of near linear length.

This addresses most of the open questions posed by our paper.

1.4 Related Works

Streaming Algorithms

The study of streaming algorithms began with the work Morris [42] on approximate counting; a rigorous analysis was given later in [20]. Later works introduced other classic problems in streaming, including heavy hitters [9], ℓ_p approximation [2, 41, 36], and finding a nonzero entry in a vector (for turnstile algorithms) [41].

Many works, for example [22, 11, 39, 3], consider the problem of processing noisy data sets using streaming algorithms. [22] shows a memory lower bound for learning a parity function with noisy samples of random linear equations.

However, this type of noise is quite different from our treatment. In these problems, noise is inherent in the way samples are generated, whereas we are investigating noise that occurs in the communication process.

Error Correcting Codes

Our result can be viewed through the lens of low-space noise resilient one-way communication.

Noise resilient one-way communication, in other words, error correction [51, 35], is one of the most well-studied problems in computer science. One specific line of works related to our result is that of *locally decodable codes*. Locally decodable codes [52] can be viewed as a low-time and low-space version of error-correcting codes, where the goal is to learn a single bit of the original message. By constrast, for us, the decoder must be able to piece together any function of the input that is noiselessly computable in low space, with the tradeoff that the decoder accesses the entire encoding via a stream rather than via query access. Local decodable codes have been constructed in a variety of different parameter regimes, including constant query [16, 15] and near 1 rate [38]. In our work, we will use Reed-Muller codes [43, 46] that achieve polylog(n) query complexity and slightly super-linear block length.

Another line of works related to ours is that of streaming algorithms for local testing of codes [47, 40]. However, this direction was rendered moot by the recent discovery of locally testable codes with constant rate, distance, and locality [14].

Coding for Interactive Communication

The analogue of noise resilient one-way communication in the interactive setting, where Alice and Bob have inputs x, y and engage in a protocol to compute some $f(x, y)$ while being resilient to error, comprises the study of *coding for interactive communication*. Interactive coding was studied starting with the seminal works of Schulman [48, 49, 50] and continuing in a prolific sequence of followup works, including [8, 6, 4, 5, 34, 7, 13, 25, 24, 17, 28, 26, 19, 30, 31]. We refer the reader to an excellent survey by Gelles [23] for an extensive list of related work.

The recent work of [18] studies the space complexity of interactive coding schemes. Their main result is an interactive coding scheme that preserves the space complexity of the original noiseless protocol up to a $\log(n)$ factor, where n is the length of the protocol. Their result

can be viewed as the interactive analogue of ours, where their noiseless and noise resilient protocols are both interactive. We remark that their techniques are quite different than ours and do not apply to our setting, however, since their approach crucially relies on the communication of feedback from Bob to Alice.

2 Overview of Techniques

Consider the task of computing a linear function $f(x) = \sum_{i\in[n]} g_i(x_i)$. In a noiseless setting, one can compute $f(x)$ by tracking the partial sum $\sum_{1\le i\le n'} g_i(x_i)$ and updating it with $g_{n'+1}(x_{n'+1})$ upon receiving the $(n'+1)$'th bit of x.

As a first attempt to create a noise resilient version of this algorithm, one might consider the input stream $\mathsf{enc}(x) = \mathsf{LDC}(x)$, where LDC is a locally decodable code. In short, a locally decodable code is a noise resilient encoding of x such that in order to decode any specific bit of x, say x_i, the decoder simply has to read polylog(n) (randomized) bits of $\mathsf{LDC}(x)$. In a streaming setting, the decoder can record only these polylog(n) bits into memory, then run the decoding algorithm to determine x_i after all such query bits have been seen.

However, it's not clear that this local decoding property extends to arbitrary linear functions.[1] If we attempt to simultaneously decode all bits x_i from $\mathsf{LDC}(x)$ using a separate query set Q_i for each index, the issue is that we will need to track $\Omega(n)$ queries simultaneously since we cannot ensure that the local decoding for any index finishes before another: the query sets Q_i are typically randomized so that any bit of the codeword is equally likely to belong to the query set for any particular index.

A second attempt is to send $\mathsf{LDC}(x)$ n times, with the intent that Bob locally decodes x_i in the i'th $\mathsf{LDC}(x)$ and computes $f(x)$ by tracking the partial sum $\sum_{1\le i\le n'} g_i(x_i)$ at the end of the i'th chunk. Now, the only space requirements are the space required to store this partial sum and the space required to locally decode x_i in each $\mathsf{LDC}(x)$, which is polylog(n). However, the adversary can simply corrupt the first $\mathsf{LDC}(x)$ so that Bob mis-decodes x_1, thereby corrupting his final output, and ultimately, this approach is not so different from just sending the bits $x_1 \dots x_n$ in order.

What Bob would like to do is randomize which x_i he is computing from a given copy of $\mathsf{LDC}(x)$ so that the adversary does not know. This crucially uses the property that LDC's allow Bob to decode *any* x_i not known to the encoder or adversary. Then, if he computes each x_i many times and takes the majority, he will learn x_i correctly with high probability, regardless of the adversary's corruption process. Since the adversary does not know in advance which chunks Bob is computing x_i in, she cannot target corruptions towards a specific x_i. However, this runs into an issue with the amount of space Bob requires. Since he computes each x_i many times throughout the protocol and only takes the majority when he has finished these computations, he needs to track his current guess for its value through (almost) the entire protocol. When each x_i was computed only once, Bob could just add $g_i(x_i)$ to the running sum, but now he needs to track each x_i individually between the first and last time he queries it, for a total of $\Omega(n)$ bits of space.

The crux of our construction is an organized system for aggregating decoding attempts in space much smaller than storing all n attempts simultaneously.

[1] The smallest local decodable code we know of that supports local decoding to any arbitrary linear function is an extension of the Hadamard code, which has block length exponential in n. This will not be a satisfactory solution for us because (a) we would like encodings that are efficiently computable, and (b) keeping track of which bit of the stream one is receiving takes n memory.

A recursive computation

Let $r = \text{polylog}(n)$. Suppose that we already know how to compute linear functions on inputs of size n/r in space $s_{n/r}$, that is resilient to $\varepsilon_{n/r}$ fraction of errors in the stream of $M_{n/r}$ copies of $\mathsf{LDC}(x)$. We will use this to build a scheme for computing linear functions on inputs of size n. Note that the base case is given by computing linear functions on a single bit, which can be done by local decoding on a single $\mathsf{LDC}(x)$.

To compute $f(x) = \sum_{i\in[n]} g_i(x_i)$, we just need to compute $f_1(x) = \sum_{i\in[n/r]} g_i(x_i)$, $f_2(x) = \sum_{i\in[n/r+1,2n/r]} g_i(x_i)$, $\ldots$, $f_r(x) = \sum_{i\in[n-n/r+1,n]} g_i(x_i)$. Each sub-computation can individually be done with the guarantee that if $< \varepsilon_{n/r}$ fraction of the stream is corrupted, then the function is computed correctly.

Consider a stream consisting of $\ell \cdot r \cdot M_{n/r}$ copies of $\mathsf{LDC}(x)$ (you can think of $\ell = \log^2 n$). We split up this stream into ℓ chunks each consisting of r blocks of $M_{n/r}$ $\mathsf{LDC}(x)$'s. In each of the ℓ chunks, we will assign the r blocks of $M_{n/r}$ LDC's to the computation of a random permutation of $f_1(x), \ldots, f_r(x)$. Throughout the algorithm, we will keep track of all the outputs from each of the ℓ sub-computations for each of $f_1(x), \ldots, f_r(x)$. At the end, we can take a majority vote for each of the $f_j(x)$. We illustrate the recursion process in Figure 1.

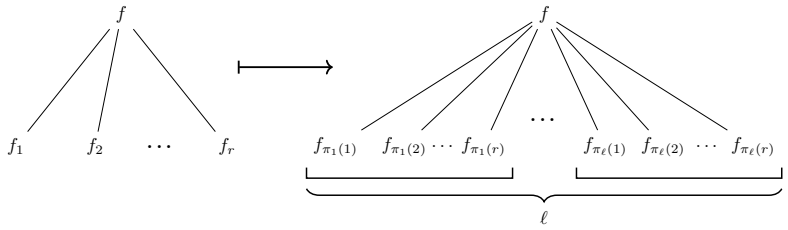


Figure 1 In our transformation, each sub-function is computed ℓ times instead of 1, and in each of the ℓ chunks, the sub-functions f_i are computed in according to a random permutation π.

Because of the way we've randomly assigned blocks to $f_1, \ldots, f_r$, the adversary cannot concentrate her attack on the computation of any specific f_i. More precisely, we can show that the amount of error present in the blocks corresponding to the computation of f_k is (roughly) proportional to the total error present. Then, as long as at least half the blocks corresponding to f_k have $< \varepsilon_{n/r}$ fraction of errors, we have that the majority vote for each of $f_1, \ldots, f_r$ will be correct. In total, this means that our algorithm for computing $f(x)$ is resilient to $\frac{\varepsilon_{n/r}}{2}$ fraction of errors.

Unfortunately, this factor of 2 loss in the error resilience in each iteration of the recursion will prove to be problematic, as we can no longer guarantee a positive error resilience when n gets large.

Decoding with confidences

In the above analysis, our worst case error distribution was that where just under one-half of the chunks have $\varepsilon_{n/r}$ error while the rest are error-free. In this case, the chunks with $\varepsilon_{n/r}$ error will produce a wrong output, while the error-free ones will produce a correct output.

However, we notice that in certain local decoding constructions (for us, we will use a Reed-Muller code), one can actually obtain an estimate of how much error there was in the codeword based on the inconsistency of the queries. In particular, in chunks where there were $\varepsilon_{n/r}$ errors, even though we will obtain a wrong answer, we can also see that there were a lot of errors in the codeword, and thereby be less sure of the obtained answer.

This motivates our definition of *local decoding with confidences*: in addition to outputting an answer, the decoding algorithm should also output a *confidence* $c \in [0, 1]$ indicating how confident it is that the answer is correct. A confidence of 0 indicates that it's not sure at all: the fraction of errors seen was near the threshold ε_1; whereas a confidence of 1 indicates that it's quite sure: either the answer is right and no errors occurred, or at least $2\varepsilon_1$ errors were required to obtain a wrong answer with confidence 1. Another way to view the confidence is as an estimate of how much error was present in the total codeword (how far the actual error is from ε_1).

Now, when we perform the recursive step to obtain a noise resilient algorithm for f from $f_1, \ldots, f_r$, in addition to recording the outputted answer from each chunk, we will also record the associated confidence. Suppose that the ℓ answers and confidences obtained for f_k were $(\hat{q}^{(1)}, \hat{c}^{(1)}), \ldots, (\hat{q}^{(\ell)}, \hat{c}^{(\ell)})$. Then, the weighted majority vote of these answers is defined to be the $\hat{q}$ with the largest total confidence. We can show that as long as the cumulative fractional error c_n is at most $c_{n/r}$, we're guaranteed that the correct answer has more the highest total confidence.

We remark that one does not have to store all ℓ pairs $(\hat{q}^{(j)}, \hat{c}^{(j)})$. Rather, we can keep track of a most likely q and associated confidence c. To update (q, c) with a new pair $(\hat{q}, \hat{c})$, one can update c by $\pm\hat{c}$ ($+\hat{c}$ if $\hat{q} = q$, and $-\hat{c}$ otherwise). If the resulting confidence c is less than 0, then it means that the answer $\hat{q}$ is supported by less overall error, so we set our most likely q to be $\hat{q}$ and flip the sign of c.

Now, we analyze the space and communication complexity of the resulting algorithm. The space required is simply $s_{n/r}$, the space required to do a size n/r sub-computation, plus the space required to store a pair (q, c) for each of $f_1, \ldots, f_r$. Overall, this means that for each layer of the recursion, we are only gaining an additive factor of $r \cdot (|q| + |c|)$ space overhead, so the total space overhead is at most polylogarithmic in n. As for the size of the resulting stream, we see that the number of $\mathsf{LDC}(x)$'s is $\approx (r \cdot \ell)^{\log_r n}$,[2] which is polynomial in n.

Computing non-linear functions

Our algorithm so far has only captured functions f that are splittable into r sub-computations whose computations do not depend on each other. However, for a general function, this may not be the case. Indeed, for many functions, the computations must be done *sequentially*, where later computations can only be done once earlier computations have finished. We may split a general function f into r sub-functions where $q_1 := f_1(x[1 : n/r], \emptyset)$ is the state of the algorithm after n/r bits of the stream, $q_2 := f_2(x[n/r + 1 : 2n/r], q_1)$ is the state of the algorithm after receiving the next n/r bits of the stream, and so on. The difference from the linear function case is that to compute f_j, we must have the correct output of f_{j-1}.

In order to handle such sequential computations, we modify our above algorithm as follows. In the recursive layer computing f from functions $f_1, \ldots, f_r$, we always compute f_j from the starting state q_{j-1} which is our current best guess for the output of f_{j-1}. This value of q_{j-1} may not always be correct, in which case the computation done for f_j will be doomed.

As with our algorithm for linear functions, after every chunk, we update the most likely state q_j and confidence c_j for the sub-function that was computed in that chunk. However, when a state q_j changes, this also means that computations done for $f_{j+1} \ldots f_n$ were based

[2] In our actual protocol, we will repeat the procedure $\log^2 n$ times and take a final majority vote, so the number of $\mathsf{LDC}(x)$'s used will be $(r \cdot \ell)^{\log_r n} \cdot \log^2 n$.

on wrong information. In the case of linear functions, the computations are independent so this does not matter, but for sequential functions, we have to discard the guesses $q_{j+1} \dots q_n$. Formally, we set these states to $\emptyset$ and set the corresponding confidences all to 0.

Because each sub-function q_j is only computed usefully when q_{j-1} is correct and also the value of q_j is frequently discarded, it is less clear that over many iterations the q_j's will converge to the correct values. Nevertheless, we prove in Section 6 that over enough iterations, the guesses q_j will eventually all be correct as long as there are not too many errors in the stream.

Outline

The rest of this paper is organized as follows. In Section 3, we start with some preliminary theorems about error-correcting codes. In Section 4, we define our notion of local decoding with confidences and prove that Reed-Muller codes satisfy this property. Then, in Sections 5 and 6, we present our noise resilient streaming algorithms for linear and general (sequential) functions, respectively.

3 Preliminaries

Notation

- The function log is in base 2 unless otherwise specified.
- The set $[n]$ denotes the integers $1 \dots n$.
- We use $x[i:j]$ to denote the i'th to j'th bits of x inclusive. We use $x(i:j]$ to refer to the $(i+1)$'th to j'th bit inclusive.
- We use e as a variable rather than the universal constant e.

3.1 Error-Correcting Codes

We begin with some results about error-correcting codes. The first is the existence of a relative distance $\frac{1}{2}$ binary code.

▶ **Theorem 4** ([33]). *For all $\epsilon > 0$, there exists an explicit error-correcting code $\mathsf{ECC}_\epsilon = \{\mathsf{ECC}_{\epsilon,n} : \{0,1\}^n \to \{0,1\}^m\}_{n \in \mathbb{N}}$ with relative distance $\frac{1}{2} - \epsilon$ and $m = O_\epsilon(n)$, and a $\mathrm{poly}_\epsilon(n)$-time decoding algorithm $\mathsf{DEC}_\epsilon : \{0,1\}^m \to \{0,1\}^n$, such that for any $x \in \{0,1\}^n$ and $w \in \{0,1\}^m$ satisfying $\Delta(\mathsf{ECC}_\varepsilon(x), w) < \left(\frac{1}{4} - \frac{\varepsilon}{4}\right) \cdot m$, it holds that $x = \mathsf{DEC}_\varepsilon(w)$.*

Our second theorem is about the efficient decoding of Reed-Solomon codes [45].

▶ **Theorem 5** (Berlekamp-Welch; see e.g. [27]). *Given a collection of tuples $(\alpha_1, v_1), \dots, (\alpha_n, v_n)$ where $\alpha_i, v_i \in \mathbb{F}_q$ and a parameter $d \leq n$, there is an algorithm running in $\mathrm{poly}(n, \log q)$ time that outputs a polynomial $g(\alpha) \in \mathbb{F}_q[\alpha]$ of degree at most $d-1$ such that $\Delta((g(\alpha_i))_{i \in [n]}, (v_i)_{i \in [n]}) < \frac{n-d+1}{2}$.*

Finally, we recall generalized minimum distance decoding, which allows us to decode concatenated codes efficiently.

▶ **Theorem 6** (GMD Decoding [21, 32]). *Given two codes, $N_{outer} : \{0,1\}^k \to \Sigma^n$ of distance $1 - \varepsilon_{outer}$ with decoding time T_{outer}, and $N_{inner} : \Sigma \to \{0,1\}^m$ of distance $\frac{1}{2} - \varepsilon_{inner}$ with decoding time D_{inner}, there is an algorithm running in time $\mathrm{poly}(D_{inner}, D_{inner})$ that on input $w \in \{0,1\}^{n \cdot m}$ outputs $x \in \{0,1\}^k$ such that $\Delta(w, N_{outer} \circ N_{inner}(x)) < \frac{(1-\varepsilon_{outer})(\frac{1}{2}-\varepsilon_{inner})}{2}$.*

4 Locally Decoding with Confidences

We now recall the definition of *locally decodable codes*. For us, besides just correctness of local decoding when the distance to a codeword is small, we require that the decoder output a *confidence* indicating how much error it witnessed (less error means higher confidence).

▶ **Theorem 7** (Locally Decodable Code). *For any $\varepsilon > 0$ and $d = d(n) = \log^{1/\delta} n$, there is a code $\mathsf{LDC} : \{0,1\}^n \to \{0,1\}^{N(n)}$ where $N(n) = O_\varepsilon(n^{1+\delta})$ that satisfies the following properties:*

- **Distance:** *For any $x \neq y \in \{0,1\}^n$, it holds that $\Delta(\mathsf{LDC}(x), \mathsf{LDC}(y)) \geq \left(\frac{1}{2} - \varepsilon\right) \cdot N$.*
- **Locally Decoding with Confidence:** *For any index $i \in [n]$ and any word $w \in \{0,1\}^N$, there exists a randomized algorithm $\mathcal{D}_i$ that reads $Q = O_\varepsilon(\log^{2/\delta} n)$ bits of w, runs in $O_\varepsilon(n^{O(1/\delta)})$ time, and outputs a bit $\hat{x}_i$ along with confidence $\mathsf{conf} \in [0,1]$, satisfying that*
  - *For any $x \in \{0,1\}^n$ such that $\Delta(w, \mathsf{LDC}(x)) = \left(\frac{1}{4} - 2\varepsilon - e\right) \cdot N$ where $e \geq 0$, it holds that*

    $$\hat{x}_i = x_i \qquad \text{and} \qquad \mathsf{conf} > e$$

    with probability at least $1 - \exp\left(-\log^2 n\right)$.
  - *For any $x \in \{0,1\}^n$ such that $\Delta(w, \mathsf{LDC}(x)) = \left(\frac{1}{4} - 2\varepsilon + e\right) \cdot N$ where $e > 0$, it holds that*

    $$\hat{x}_i = x_i \qquad \text{or} \qquad \mathsf{conf} < e$$

    with probability at least $1 - \exp\left(-\log^2 n\right)$.

  Furthermore, the confidence conf can be represented as a rational number with denominator $4Q$.

Proof. This proof is omitted for the conference version. ◀

5 Noise Resilient Streaming for Linear Algorithms

Our first result is a noise resilient conversion for linear streaming algorithms A. We begin by recalling the definition of a linear streaming algorithm.

▶ **Definition 2** (Linear Streaming Algorithms). *A linear streaming algorithm $A : \{0,1\}^n \to \mathbb{F}_q$ is described by a list of functions $g_i : \{0,1\} \to \mathbb{F}_q$ for $i \in [n]$, and computes the value $A(x) = g_1(x_1) + \ldots + g_n(x_n)$ (where addition is over $\mathbb{F}_q$) by tracking the partial sum $g_1(x_1) + \ldots + g_i(x_i)$ upon receiving the i'th bit.*

For such algorithms, we describe a noise resilient conversion that incurs quadratic blow-up in communication complexity.

▶ **Theorem 3.** *Fix $\varepsilon, \delta > 0$. There is a function $\mathsf{enc} : X \subseteq \{0,1\}^n \to \{0,1\}^m$ where $m = O_\varepsilon(n^{2+\delta})$ and an explicit transformation B such that the following holds: For any linear streaming algorithm A that takes as input $x \in X \subseteq \{0,1\}^n$ as a stream, runs in space s and time t, and outputs $A(x)$, the algorithm $B_A = B(A)$ takes as input $z \in \{0,1\}^m$ as a stream, runs in space $s \cdot O_\varepsilon((\log n)^{O(1/\delta)})$ and time $m \cdot O_{\varepsilon,\delta}\left(1 + \frac{t}{n^2}\right)$, and satisfies that whenever $\Delta(z, \mathsf{enc}(x)) < \left(\frac{1}{4} - \varepsilon\right) \cdot m$ then $B_A(z)$ outputs $A(x)$ with probability $\geq 1 - 2^{O_\varepsilon(\log^2 n)}$.*

5.1 Statement of Algorithm

Let $\varepsilon > 0$. Throughout the section let $r = \log^{1/\delta} n$ and define $\ell := \log^2 n$. We also assume for simplicity that $\log_r n$ is an integer. We will also assume that $n > \max\{\exp(\exp(8\delta/\varepsilon)), \exp(1/\varepsilon)\}$ is sufficiently large, so that also $\ell < n$.

We begin by specifying the encoding $\mathsf{enc}(x)$ that Alice sends in the stream.

▶ **Definition 8** ($\mathsf{enc}(x)$)**.** *Let $\mathsf{LDC}(x)$ be a locally decodable code with $N(n) = |\mathsf{LDC}(x)| = O_\varepsilon(n^{1+\delta})$ and query complexity $Q = O_\varepsilon(\log^{2/\delta} n)$ satisfying the guarantees of Theorem 7 for $\varepsilon/8$. Then Alice sends $\mathsf{enc}(x) := \mathsf{LDC}(x)^{\log^2 n \cdot M_n}$ where $M_n = (r \cdot \ell)^{\log_r n}$ (that is, $\mathsf{enc}(n)$ is $\log^2 n \cdot M_n$ copies of $\mathsf{LDC}(x)$). In particular, $m = |\mathsf{enc}(x)| = (r \cdot \ell)^{\log_r n} \cdot \log^2 n \cdot N(n) = O_\varepsilon(n^{2+4\delta})$.*

Throughout this section, we use the notation $A_{i,j}(x)$ to denote the quantity $g(x_{i+1}) + \ldots + g_j(x_j)$. Then, $A(x) = A_{0,n}(x)$. We define $N_{j-i} := (r \cdot \ell)^{\log_r(j-i)} \cdot N(n)$ which, as we'll see in the description below, represent the number of bits read by the algorithm to approximate $A_{i,j}(x)$.

We describe Bob's streaming algorithm $B_A := B(A)$. Before stating it formally, we provide a high-level description.

Description

Let z be the incoming stream that Bob receives. At a high level, the algorithm B_A runs by generating many guesses for $A(x)$ along with confidences indicating how likely that guess was correct (depending on how much error is witnessed while generating the guess). At the end, the guess that has the most cumulative confidence is outputted.

For i, j where $j - i$ is an integer power of r, we define the algorithm $\mathsf{estA}(i, j)$ that takes in two input two indices $i, j \in [n]$. It reads N_{j-i} many bits of z after which it outputs a guess for $A_{i,j}(x)$ along with a confidence. In particular, $\mathsf{estA}(0, n)$ outputs a guess for $A(x)$ along with a confidence. By running $\mathsf{estA}(0, n)$ many times and aggregating the guesses weighted by the confidences, we obtain a final output.

To compute $\mathsf{estA}(i, j)$, we break down the interval $(i : j]$ into r subintervals $(i_0 : i_1], (i_1 : i_2], \ldots, (i_{r-1} : i_r]$ where $i_a = i + a \cdot \frac{j-i}{r}$. We then recursively formulate guesses for each $A\left(x\left(i_{a-1} : i_a\right]\right)$ over many iterations by calling $\mathsf{estA}(i_{a-1}, i_{a)}$ each many times. The choice of a in each iteration must be randomized, so that the adversary cannot attack any single choice of a. More specifically, we will split up the N_{j-i} length input stream into ℓ chunks, each of length N_{j-i}/ℓ bits. Each chunk is split into r sections, each of size $N_{j-i}/(\ell \cdot r) = N_{(j-i)/r}$. In each chunk, we pick a random permutation $(\pi_1, \pi_2, \ldots, \pi_r)$ of $[r]$. In the a'th section of the chunk, we will compute on the subproblem corresponding to interval $(i_{\pi_a - 1} : i_{\pi_a}]$ by calling $\mathsf{estA}\left(i_{\pi_a - 1}, i_{\pi_a}\right)$.

Whenever a recursive call to $\mathsf{estA}\left(i_{a-1}, i_a\right)$ is completed, outputting $(\hat{q}, \hat{c})$, then: if $q_a = \hat{q}$ then the confidence c_a is increased by $\hat{c}$, and if $q_a \neq \hat{q}$ then the confidence c_a is decreased by $\hat{c}$. If the confidence c_a becomes negative, then we are more sure that the correct state is $\hat{q}$ rather than q_a: then q_a is replaced by $\hat{q}$ and the confidence is negated (so that it's positive).

5.2 Proof of Theorem 3

We begin by proving the stated claims about the communication, time, and space complexities, and then proceed to prove correctness.

Algorithm 1 Bob's Noise Resilient Algorithm B_A.

input $n \in \mathbb{N}$ and stream $z \in \{0,1\}^m$.
function $\mathsf{estA}(i,j)$ ▷compute the state of A starting at state q between steps i and j
  if $j = i+1$ **then**
    Read the next $|\mathsf{LDC}(x)|$ bits of the stream y.
    Using Theorem 7 compute guess $\hat{b}$ for $x[j]$ and confidence $\hat{c}$ in $O_\varepsilon\left((\log n)^{O(1/\delta)}\right)$ bits of space.
    return $g_i(\hat{b}), c$.
  else
    Let $i_0 = i, i_1 = i + \frac{j-i}{r}, i_2 = i + \frac{2(j-i)}{r}, \ldots, i_r = j$.
    Initialize a list of pairs $(q_1, c_1), \ldots (q_r, c_r)$ each to $(\emptyset, 0)$. ▷cumulative best guesses and confidences
    for ℓ iterations **do**
      Let $\pi_1 \ldots \pi_r$ be a random permutation of $[r]$.
      for $a \in [r]$ **do**
        Compute $(\hat{q}, \hat{c}) \leftarrow \mathsf{estA}(i_{\pi_a - 1}, i_{\pi_a})$.
        if $\hat{q} = q_{\pi_a}$ **then** ▷update the confidence on $\mathsf{estA}(i_{\pi_a}, i_{\pi_j + 1}, q_{\pi_a - 1})$
          $c_{\pi_a} \leftarrow c_{\pi_a} + \hat{c}$
        else
          $c_{\pi_a} \leftarrow c_{\pi_a} - \hat{c}$
          if $c_{\pi_a} < 0$ **then** ▷if the guess changes, flip its confidence
            $q_{\pi_a} \leftarrow \hat{q}$ and $c_{\pi_a} \leftarrow -c_{\pi_a}$.
    return $q_1 + \ldots + q_r, \min(c_1 \ldots c_r)/\ell$

Initialize a pair (q, c) to $(\emptyset, 0)$.
for $\log^2 n$ iterations **do** ▷amplification step
  Let $(\hat{q}, \hat{c}) \leftarrow \mathsf{estA}(0, n)$
  if $\hat{q} = q$ **then**
    $c \leftarrow c + \hat{c}$
  else
    $c \leftarrow c - \hat{c}$
    if $c < 0$ **then**
      $q \leftarrow \hat{q}$ and $c \leftarrow -c$.
output q. ▷output $A(x)$

5.2.1 Algorithmic Complexity

The proofs of communication, time, and space complexities are omitted for the conference version.

5.2.2 Correctness

We now show correctness. Formally, correctness is shown by the following lemma.

▶ **Lemma 9.** *When at most* $\frac{1}{4} - \varepsilon$ *fraction of the stream* $\mathsf{enc}(x)$ *is corrupted, Algorithm 1 outputs* $A(x)$ *with probability* $\geq 1 - \exp\left(-\varepsilon^2 \log^2 n/32\right)$.

We prove the following statement which will easily imply Lemma 9. For a given i, j and associated string that is read by the algorithm in the computation, let the random variable $(q(i,j), c(i,j))$ denote the state and confidence after the computation $\mathsf{estA}(i,j)$. We

define the *signed confidence*, denoted $c^{\pm}(i,j)$, to be defined as $+c(i,j)$ if $q(i,j) = A_{i,j}(x)$ and $-c(i,j)$ otherwise. For intuition, the more positive $c^{\pm}(i,j)$ is, the more correct with higher confidence the output $(q(i,j), c(i,j))$ is, whereas the more negative $c_{\pm}(i,j)$ is, the more $q(i,j)$ is incorrect with high confidence. So, $c^{\pm}(i,j)$ gives us a scaled measure of how correct the output of $\mathsf{estA}(i,j)$ is.

▶ **Lemma 10.** *For any $0 \leq i < j \leq n$ and $e \in \mathbb{R}$, given $\leq \frac{1}{4} - \frac{\varepsilon}{4} - \frac{\varepsilon}{4} \cdot \frac{\log(j-i)}{\log n} - e$ fraction of corruptions in the bits read by the computation of $\mathsf{estA}(i,j)$, we have $\mathbb{E}[c^{\pm}(i,j)] > e$.*

Proof. The proof of this lemma is omitted for the conference version. ◀

Proof of Lemma 9. In the special case where $i = 0, j = n$ and there are $\leq \frac{1}{4} - \frac{\varepsilon}{2} - e < \frac{1}{4} - \frac{\varepsilon}{4} - \frac{\varepsilon}{4} \cdot \frac{\log(j-i)}{r} - e$ errors (where the inequality follows because $\log n < \varepsilon \cdot r$), by Lemma 10 we have that $\mathbb{E}[c^{\pm}(0,n)] > e$.

Then, the final amplification step of the protocol computes the pair $(q(0,n), c(0,n))$ for $\log^2 n$ times, where in each chunk i we denote the fraction of error to be $\left(\frac{1}{4} - \frac{\varepsilon}{2} - e_i\right)$, where $\sum_i e_i \geq \frac{\varepsilon \log^2 n}{2}$ since we assumed the total error was $\leq \frac{1}{4} - \varepsilon$. Also, the protocol outputs $A(x)$ if and only if $\sum_i c_i^{\pm}(0,n)$ (denoting the value of $c^{\pm}(0,n)$ in the i'th chunk) is positive. By Azuma's inequality,

$$Pr\left[\sum_i c_i^{\pm}(0,n) > 0\right] \geq Pr\left[\sum_i \left(c_i^{\pm}(0,n) - e_i\right) > -\frac{\varepsilon \log^2 n}{2}\right] \geq 1 - \exp\left(-\frac{\varepsilon^2 \log^2 n}{32}\right).$$ ◀

This concludes the proof of Theorem 3.

6 Noise Resilient Streaming for General Algorithms

We now consider general streaming algorithms, whose computational may be sequential in nature. Our main result is a noise resilient conversion for deterministic streaming algorithms A. Compared to our scheme for linear algorithms from Section 5, the length of our encoding is larger: $n^{4+\delta}$ compared to $n^{2+\delta}$.

▶ **Theorem 1.** *Fix $\varepsilon, \delta > 0$. For any $X \subseteq \{0,1\}^n$, there is a noise resilient transformation consisting of an encoding $\mathsf{enc} : X \subseteq \{0,1\}^n \to \{0,1\}^m$, and a transformation of algorithms B that is $\left(\frac{1}{4} - \varepsilon\right) \cdot m$ error resilient with probability $\geq 1 - 2^{O_\varepsilon(\log^2 n)}$. Moreover, $m = O_\varepsilon(n^{4+\delta})$, and for any deterministic algorithm A on domain X that runs in space s and time t, the algorithm $B(A)$ runs in space $s \cdot O_\varepsilon\left((\log n)^{O(1/\delta)}\right)$ and time $m \cdot O_{\varepsilon,\delta}\left(1 + \frac{t}{n^2}\right)$.*

Throughout this section, we use the notation

$$A(q, \hat{x}) \in \{0,1\}^s$$

to denote the state of algorithm A when starting with the state $q \in \{0,1\}^s$ and executing on $\hat{x}$ received in a stream. By definition, there is an explicit algorithm that computes $A(q, \hat{x})$ in s space. We also use the shorthand $A(\hat{x})$ to denote $A(\emptyset, \hat{x})$. Notice that $A(x) = A(\emptyset, x)$ is simply the output state of the stream.

6.1 Statement of Algorithm

Let $\varepsilon > 0$. Throughout the section let $r = \log^{1/\delta} n$ and define $\ell := r^2 \log^4 n$. We also assume for simplicity that $\log_r n$ is an integer. We will also assume that the parameter $n > \max\{\exp(\exp(8\delta/\varepsilon)), \exp(1/\varepsilon)\}$ is sufficiently large, so that also $\ell < n$.

We begin by specifying the encoding $\mathsf{enc}(x)$ that Alice sends in the stream.

▶ **Definition 11** ($\mathsf{enc}(x)$)**.** *Let* $\mathsf{LDC}(x)$ *be a locally decodable code with* $N(n) = |\mathsf{LDC}(x)| = O_\varepsilon(n^{1+\delta})$ *and query complexity* $Q = O_\varepsilon(\log^{2/\delta} n)$ *satisfying the guarantees of Theorem 7 for* $\varepsilon/8$*. Then Alice sends* $\mathsf{enc}(x) := \mathsf{LDC}(x)^{\log^2 n \cdot M_n}$ *where* $M_n = (r \cdot \ell)^{\log_r n}$ *(that is,* $\mathsf{enc}(n)$ *is* $\log^2 n \cdot M_n$ *copies of* $\mathsf{LDC}(x)$*). In particular,* $m = |\mathsf{enc}(x)| = (r \cdot \ell)^{\log_r n} \cdot \log^2 n \cdot N(n) = O_\varepsilon(n^{4+6\delta})$.

Next, we describe Bob's streaming algorithm $B_A := B(A)$. Before stating it formally, we provide a high-level description.

Description

Let z be the incoming stream that Bob receives. At a high level, the algorithm B_A runs by generating many guesses for $A(x)$ along with confidences indicating how likely that guess was correct (depending on how much error is witnessed while generating the guess). At the end, the guess that has the most cumulative confidence is outputted.

For i, j where $j - i$ is an integer power of r, we define the algorithm $\mathsf{estA}(i, j, q)$ that has the following syntax:

- $\mathsf{estA}(i, j, q)$ takes as input two indices $i, j \in [n]$ along with a state $q \in \{0,1\}^s$ which represents a guess for the state of $A(q, x(i : j])$.
- It reads $N_{j-i} := (r \cdot \ell)^{\log_r(j-i)} \cdot N(n)$ many bits of z after which it outputs a guess for $A(q, x(i : j])$ along with a confidence.

In particular, $\mathsf{estA}(0, n, q)$ outputs a guess for $A(x)$ along with a confidence. By running $\mathsf{estA}(0, n, q)$ many times and aggregating the guesses weighted by the confidences, we obtain a final output.

At a high level, to compute $\mathsf{estA}(i, j, q)$, we break down the interval $(i : j]$ into r subintervals $(i_0 : i_1], (i_1 : i_2], \ldots, (i_{r-1} : i_r]$ where $i_a = i + a \cdot \frac{j-i}{r}$. We then recursively formulate guesses for each $A(x(1 : i_a])$ over many iterations by calling $\mathsf{estA}(i_{a-1}, i_a, q_{a-1})$ for some state q_{a-1}. The choice of a in each iteration must be randomized, so that the adversary cannot attack any single choice of a. More specifically, we will split up the N_{j-i} length input stream into ℓ chunks, each of length N_{j-i}/ℓ bits. Each chunk is split into r sections, each of size $N_{j-i}/(\ell \cdot r) = N_{(j-i)/r}$. In each chunk, we pick a random permutation $(\pi_1, \pi_2, \ldots, \pi_r)$ of $[r]$. In the a'th section of the chunk, we will compute on the subproblem corresponding to interval $(i_{\pi_a - 1} : i_{\pi_a}]$ by calling $\mathsf{estA}(i_{\pi_a - 1}, i_{\pi_a}, q_{\pi_a - 1})$.

Note that as the algorithm A is computed sequentially, computing a guess for $A(x[1 : i_a])$ requires having a starting state q_{a-1} for $A(x[1 : i_{a-1}])$. Thus, throughout the entire computation, we will keep track of the best guess and associated confidence for each of the r states $A(x[1 : i_a])$, denoted (q_a, c_a) and all initialized to $(\emptyset, 0)$.

These pairs (q_a, c_a) are maintained as follows. Whenever a recursive call to $\mathsf{estA}(i_{a-1}, i_a, q_{a-1})$ is completed, outputting $(\hat{q}, \hat{c})$, then: if $q_a = \hat{q}$ then the confidence c_a is increased by $\hat{c}$, and if $q_a \neq \hat{q}$ then the confidence c_a is decreased by $\hat{c}$. If the confidence c_a becomes negative, then we are more sure that the correct state is $\hat{q}$ rather than q_a: then q_a is replaced by $\hat{q}$ and the confidence is negated (so that it's positive). In this last case where q_a is replaced, we have no more reason to believe that further computations of $q_{a'}$, $a' > a$, are correct since they all depended on q_a, so we erase all such pairs $(q_{a'}, c_{a'})$ and reset them to $(\emptyset, 0)$. A key point in our analysis is to understand why this does not cause the error probability to accumulate.

Algorithm 2 Bob's Noise Resilient Algorithm B_A

input $n \in \mathbb{N}$ and stream $z \in \{0,1\}^m$.
function $\mathsf{estA}(i, j, q)$ ▷compute the state of A starting at state q between steps i and j
  if $j = i + 1$ **then**
    Read the next $|\mathsf{LDC}(x)|$ bits of the stream y.
    Using Theorem 7 compute guess $\hat{b}$ for $x[j]$ and confidence $\hat{c}$ in $s \cdot O_\varepsilon\left((\log n)^{O(1/\delta)}\right)$ bits of space.
    return $A(q, \hat{b}), c$.
  else
    Let $i_0 = i, i_1 = i + \frac{j-i}{r}, i_2 = i + \frac{2(j-i)}{r}, \ldots, i_r = j$.
    Initialize a list of pairs $(q_1, c_1), \ldots (q_r, c_r)$ each to $(\emptyset, 0)$. ▷cumulative best guesses and confidences
    for ℓ iterations **do**
      Let $\pi_1 \ldots \pi_r$ be a random permutation of $[r]$.
      Set $(q'_1, c'_1) = (q_1, c_1), \ldots (q'_r, c'_r) = (q_r, c_r)$.
      for $a \in [r]$ **do**
        Compute $(\hat{q}, \hat{c}) \leftarrow \mathsf{estA}(i_{\pi_a - 1}, i_{\pi_a}, q'_{\pi_a - 1})$ where $q'_0 := q$.
        if $\hat{q} = q'_{\pi_a}$ **then** ▷update the confidence on $\mathsf{estA}(i_{\pi_a}, i_{\pi_j + 1}, q_{\pi_a - 1})$
          $c_{\pi_a} \leftarrow c'_{\pi_a} + \hat{c}$
        else
          $c_{\pi_a} \leftarrow c'_{\pi_a} - \hat{c}$
          if $c_{\pi_a} < 0$ **then** ▷if the guess changes, reset its confidence
            $q_{\pi_a} \leftarrow \hat{q}$ and $c_{\pi_a} \leftarrow -c_{\pi_a}$.
      if some q_a was changed from its value at the beginning of the chunk **then**
        For all $i > a$, set $(q_i, c_i) \leftarrow (\emptyset, 0)$. ▷reset state and confidence for states with $i > a$
    return $q_r, \min(c_1 \ldots c_r)/\ell$

Initialize a pair (q, c) to $(\emptyset, 0)$.
for $\log^2 n$ iterations **do** ▷amplification step
  Let $(\hat{q}, \hat{c}) \leftarrow \mathsf{estA}(0, n, \emptyset)$
  if $\hat{q} = q$ **then**
    $c \leftarrow c + \hat{c}$
  else
    $c \leftarrow c - \hat{c}$
    if $c < 0$ **then**
      $q \leftarrow \hat{q}$ and $c \leftarrow -c$.
output q. ▷output $A(x)$

6.2 Proof of Theorem 1

The proofs that the algorithm satisfies the required space, computational and communication guarantees are essentially the same as in Section 5, but we reproduce them for completeness. The proof of correctness is somewhat more complicated but also follows the same general outline.

6.2.1 Algorithmic Complexity

The proofs of communication, time, and space complexities are omitted for the conference version.

6.2.2 Correctness

Next, we show correctness. Formally, correctness is shown by the following lemma.

▶ **Lemma 12.** *When at most* $\frac{1}{4} - \varepsilon$ *fraction of the stream* $\mathsf{enc}(x)$ *is corrupted, Algorithm 2 outputs* $A(x)$ *with probability* $\geq 1 - \exp\left(-\varepsilon^2 \log^2 n/32\right)$.

We prove the following statement which will easily imply Lemma 12. For a given i, j, q and associated string that is read by the algorithm in the computation, let the random variable $(q(i,j,q), c(i,j,q))$ denote the state and confidence after the computation $\mathsf{estA}(i,j,q)$. We define the *signed confidence*, denoted $c^{\pm}(i,j,q)$, to be defined as $+c(i,j,q)$ if $q(i,j,q) = A(q, x(i:j])$ and $-c(i,j,q)$ otherwise. For intuition, the more positive $c^{\pm}(i,j,q)$ is, the more correct with higher confidence the output $(q(i,j,q), c(i,j,q))$ is, whereas the more negative $c_{\pm}(i,j,q)$ is, the more $q(i,j,q)$ is incorrect with high confidence. So, $c^{\pm}(i,j,q)$ gives us a scaled measure of how correct the output of $\mathsf{estA}(i,j,q)$ is.

▶ **Lemma 13.** *For any* $0 \leq i < j \leq n$, $q \in \{0,1\}^s$, *and* $e \in \mathbb{R}$, *given* $\leq \frac{1}{4} - \frac{\varepsilon}{4} - \frac{\varepsilon}{4} \cdot \frac{\log(j-i)}{\log n} - e$ *fraction of corruptions in the bits read by the computation of* $\mathsf{estA}(i,j,q)$, *we have* $\mathbb{E}[c^{\pm}(i,j,q)] > e$.

Proof. The proof of this lemma is omitted for the conference version. ◀

Proof of Lemma 12. In the special case where $i = 0, j = n, q = \emptyset$ and there are $\leq \frac{1}{4} - \frac{\varepsilon}{2} - e < \frac{1}{4} - \frac{\varepsilon}{4} - \frac{\varepsilon}{4} \cdot \frac{\log(j-i)}{r} - e$ errors (where the inequality follows because $\log n < \varepsilon \cdot r$), by Lemma 13 we have that $\mathbb{E}[c^{\pm}(0,n,\emptyset)] > e$.

Then, the final amplification step of the protocol computes the pair $(q(0,n,\emptyset), c(0,n,\emptyset))$ for $\log^2 n$ times, where in each chunk i we denote the fraction of error to be $\left(\frac{1}{4} - \frac{\varepsilon}{2} - e_i\right)$, where $\sum_i e_i \geq \frac{\varepsilon \log^2 n}{2}$ since we assumed the total error was $\leq \frac{1}{4} - \varepsilon$. Also, the protocol outputs $A(x)$ if and only if $\sum_i c_i^{\pm}(0,n,\emptyset)$ (denoting the value of $c^{\pm}(0,n,\emptyset)$ in the i'th chunk) is positive. By Azuma's inequality,

$$Pr\left[\sum_i c_i^{\pm}(0,n,\emptyset) > 0\right] \geq Pr\left[\sum_i \left(c_i^{\pm}(0,n,\emptyset) - e_i\right) > -\frac{\varepsilon \log^2 n}{2}\right] \geq 1 - \exp\left(-\frac{\varepsilon^2 \log^2 n}{32}\right).$$

◀

This concludes the proof of Theorem 1.

References

1 Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. *Analyzing Graph Structure via Linear Measurements*, pages 459–467. SIAM, 2012. doi:10.1137/1.9781611973099.40.

2 Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 20–29, 1996. doi:10.1145/237814.237823.

3 Omri Ben-Eliezer, Rajesh Jayaram, David P Woodruff, and Eylon Yogev. A framework for adversarially robust streaming algorithms. *ACM Journal of the ACM (JACM)*, 69(2):1–33, 2022. doi:10.1145/3498334.

4 Zvika Brakerski and Yael Tauman Kalai. Efficient Interactive Coding against Adversarial Noise. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 160–166, 2012. doi:10.1109/FOCS.2012.22.

5 Zvika Brakerski and Moni Naor. Fast Algorithms for Interactive Coding. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 443–456, USA, 2013. Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611973105.32.

6 Mark Braverman. Towards Deterministic Tree Code Constructions. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 161–167, New York, NY, USA, 2012. Association for Computing Machinery. doi:10.1145/2090236.2090250.

7 Mark Braverman and Klim Efremenko. List and Unique Coding for Interactive Communication in the Presence of Adversarial Noise. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 236–245, Los Alamitos, CA, USA, October 2014. IEEE Computer Society. doi:10.1109/FOCS.2014.33.

8 Mark Braverman and Anup Rao. Towards Coding for Maximum Errors in Interactive Communication. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 159–166, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/1993636.1993659.

9 Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *International Colloquium on Automata, Languages, and Programming*, pages 693–703. Springer, 2002. doi:10.1007/3-540-45465-9_59.

10 Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theoretical Computer Science*, 312(1):3–15, 2004. Automata, Languages and Programming. doi:10.1016/S0304-3975(03)00400-6.

11 Jiecao Chen and Qin Zhang. Bias-aware sketches. *arXiv preprint*, 2016. arXiv:1610.07718.

12 Graham Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005. doi:10.1016/j.jalgor.2003.12.001.

13 Varsha Dani, Thomas P. Hayes, Mahnush Movahedi, Jared Saia, and Maxwell Young. Interactive Communication with Unknown Noise Rate, 2015. arXiv:1504.06316.

14 Irit Dinur, Shai Evra, Ron Livne, Alexander Lubotzky, and Shahar Mozes. Locally testable codes with constant rate, distance, and locality. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 357–374, 2022. doi:10.1145/3519935.3520024.

15 Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. *SIAM Journal on Computing*, 40(4):1154–1178, 2011. doi:10.1137/100804322.

16 Klim Efremenko. 3-query locally decodable codes of subexponential length. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 39–44, 2009. doi:10.1145/1536414.1536422.

17 Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Maximal Noise in Interactive Communication Over Erasure Channels and Channels With Feedback. *IEEE Trans. Inf. Theory*, 62(8):4575–4588, 2016. doi:10.1109/TIT.2016.2582176.

18 Klim Efremenko, Bernhard Haeupler, Yael Tauman Kalai, Gillat Kol, Nicolas Resch, and Raghuvansh R Saxena. Interactive coding with small memory. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3587–3613. SIAM, 2023. doi:10.1137/1.9781611977554.CH137.

19 Klim Efremenko, Gillat Kol, and Raghuvansh R. Saxena. Binary Interactive Error Resilience Beyond $1/8$ (or why $(1/2)^3 > 1/8$). In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 470–481, 2020. doi:10.1109/FOCS46700.2020.00051.

20 Philippe Flajolet. Approximate counting: a detailed analysis. *BIT Numerical Mathematics*, 25(1):113–134, 1985. doi:10.1007/BF01934993.

21 G. Forney. Generalized minimum distance decoding. *IEEE Transactions on Information Theory*, 12(2):125–131, 1966. doi:10.1109/TIT.1966.1053873.

22 Sumegha Garg, Pravesh K Kothari, Pengda Liu, and Ran Raz. Memory-sample lower bounds for learning parity with noise. *arXiv preprint*, 2021. arXiv:2107.02320.

23 Ran Gelles. Coding for Interactive Communication: A Survey. *Foundations and Trends® in Theoretical Computer Science*, 13:1–161, January 2017. doi:10.1561/0400000079.

24 Ran Gelles and Bernhard Haeupler. Capacity of Interactive Communication over Erasure Channels and Channels with Feedback. *SIAM Journal on Computing*, 46:1449–1472, January 2017. doi:10.1137/15M1052202.

25 Ran Gelles, Bernhard Haeupler, Gillat Kol, Noga Ron-Zewi, and Avi Wigderson. *Towards Optimal Deterministic Coding for Interactive Communication*, pages 1922–1936. SIAM, 2016. doi:10.1137/1.9781611974331.ch135.

26 Ran Gelles and Siddharth Iyer. Interactive coding resilient to an unknown number of erasures. *arXiv preprint*, 2018. arXiv:1811.02527.

27 Peter Gemmell and Madhu Sudan. Highly resilient correctors for polynomials. *Information Processing Letters*, 43(4):169–174, 1992. doi:10.1016/0020-0190(92)90195-2.

28 Mohsen Ghaffari and Bernhard Haeupler. Optimal Error Rates for Interactive Coding II: Efficiency and List Decoding. *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, December 2013. doi:10.1109/FOCS.2014.49.

29 Meghal Gupta, Venkatesan Guruswami, and Mihir Singhal. Tight bounds for stream decodable error-correcting codes, 2024. doi:10.48550/arXiv.2407.06446.

30 Meghal Gupta and Rachel Yun Zhang. Efficient interactive coding achieving optimal error resilience over the binary channel. *arXiv preprint*, 2022. doi:10.48550/arXiv.2207.01144.

31 Meghal Gupta and Rachel Yun Zhang. The Optimal Error Resilience of Interactive Communication Over Binary Channels. In *Symposium on Theory of Computing, STOC 2012, New York, NY, USA, June 20 - June 24, 2022*, STOC '22. ACM, 2022. doi:10.1145/3519935.3519985.

32 Venkat Guruswami. Error-correcting codes: Constructions and algorithms, lecture no. 11, 2006. URL: http://www.cs.washington.edu/education/courses/533/06au/.

33 Venkatesan Guruswami and Madhu Sudan. List decoding algorithms for certain concatenated codes. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, STOC '00, pages 181–190, New York, NY, USA, 2000. Association for Computing Machinery. doi:10.1145/335305.335327.

34 Bernhard Haeupler. Interactive Channel Capacity Revisited. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 226–235, 2014. doi:10.1109/FOCS.2014.32.

35 Richard W Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.

36 Piotr Indyk and David Woodruff. Optimal approximations of the frequency moments of data streams. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 202–208, 2005. doi:10.1145/1060590.1060621.

37 William Johnson and Joram Lindenstrauss. Extensions of lipschitz maps into a hilbert space. *Contemporary Mathematics*, 26:189–206, January 1984. doi:10.1090/conm/026/737400.

38 Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. *J. ACM*, 61(5), September 2014. doi:10.1145/2629416.

39 Chaoyi Ma, Haibo Wang, Olufemi Odegbile, and Shigang Chen. Noise measurement and removal for data streaming algorithms with network applications. In *2021 IFIP Networking Conference (IFIP Networking)*, pages 1–9. IEEE, 2021. doi:10.23919/IFIPNETWORKING52078.2021.9472845.

40 Andrew McGregor, Atri Rudra, and Steve Uurtamo. Polynomial fitting of data streams with applications to codeword testing. *Symposium on Theoretical Aspects of Computer Science (STACS2011)*, 9, March 2011. doi:10.4230/LIPIcs.STACS.2011.428.

41 Morteza Monemizadeh and David P Woodruff. 1-pass relative-error lp-sampling with applications. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1143–1160. SIAM, 2010. doi:10.1137/1.9781611973075.92.

42 Robert Morris. Counting large numbers of events in small registers. *Communications of the ACM*, 21(10):840–842, 1978. doi:10.1145/359619.359627.

43 David E Muller. Application of boolean algebra to switching circuit design and to error detection. *Transactions of the IRE professional group on electronic computers*, 3:6–12, 1954. doi:10.1109/IREPGELC.1954.6499441.

44 N. Nisan. Pseudorandom generators for space-bounded computations. In *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, STOC '90, pages 204–212, New York, NY, USA, 1990. Association for Computing Machinery. doi:10.1145/100216.100242.

45 I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960. doi:10.1137/0108018.

46 Irving S Reed. A class of multiple-error-correcting codes and the decoding scheme. *IEEE Transactions on Information Theory*, 4(4):38–49, 1954. doi:10.1109/TIT.1954.1057465.

47 Atri Rudra and Steve Uurtamo. Data stream algorithms for codeword testing. In *International Colloquium on Automata, Languages, and Programming*, pages 629–640. Springer, 2010. doi:10.1007/978-3-642-14165-2_53.

48 Leonard J. Schulman. Communication on noisy channels: a coding theorem for computation. In *Proceedings., 33rd Annual Symposium on Foundations of Computer Science*, pages 724–733, 1992. doi:10.1109/SFCS.1992.267778.

49 Leonard J. Schulman. Deterministic Coding for Interactive Communication. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '93, pages 747–756, New York, NY, USA, 1993. Association for Computing Machinery. doi:10.1145/167088.167279.

50 Leonard J. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996. doi:10.1109/18.556671.

51 Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. doi:10.1002/j.1538-7305.1948.tb01338.x.

52 Sergey Yekhanin et al. Locally decodable codes. *Foundations and Trends® in Theoretical Computer Science*, 6(3):139–255, 2012. doi:10.1561/0400000030.

List Decoding Bounds for Binary Codes with Noiseless Feedback

Meghal Gupta
University of California Berkeley, CA, USA

Rachel Yun Zhang
Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

In an error-correcting code, a sender encodes a message $x \in \{0,1\}^k$ such that it is still decodable by a receiver on the other end of a noisy channel. In the setting of *error-correcting codes with feedback*, after sending each bit, the sender learns what was received at the other end and can tailor future messages accordingly.

While the unique decoding radius of feedback codes has long been known to be $\frac{1}{3}$, the list decoding capabilities of feedback codes is not well understood. In this paper, we provide the first nontrivial bounds on the list decoding radius of feedback codes for lists of size ℓ. For $\ell = 2$, we fully determine the 2-list decoding radius to be $\frac{3}{7}$. For larger values of ℓ, we show an upper bound of $\frac{1}{2} - \frac{1}{2^{\ell+2}-2}$, and show that the same techniques for the $\ell = 2$ case cannot match this upper bound in general.

2012 ACM Subject Classification Mathematics of computing → Coding theory

Keywords and phrases error-correcting codes, feedback, list decoding

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.60

Related Version *Full Version*: https://arxiv.org/abs/2410.01951

Funding *Meghal Gupta*: Supported by a NSF GFRP Fellowship.
Rachel Yun Zhang: Supported by NSF Graduate Research Fellowship 2141064. Supported in part by DARPA under Agreement No. HR00112020023 and by an NSF grant CNS-2154149.

1 Introduction

Consider the problem of error-correcting codes with feedback [3]. Alice wishes to communicate a message $x \in \{0,1\}^k$ to Bob by transmitting a sequence of n bits over a noisy channel. After sending each bit, she receives *feedback* through a noiseless channel, telling her what bit Bob actually received. She can use this feedback to adaptively decide her next bit of communication, potentially allowing for improvement over classical error-correcting codes. As in the classical setting, the goal is to communicate in a way such that even if some fraction r of Alice's bits are adversarially corrupted, Bob can still correctly determine x.

For classical error-correcting codes *without* feedback, the Plotkin bound [26] states that Bob cannot uniquely decode x if more than $\frac{1}{4}$ of the communication is adversarially corrupted. In other words, the unique decoding radius for binary error-correcting codes is $\frac{1}{4}$.[1] The corresponding question for codes with feedback was first posed by Berlekamp in his 1964 thesis, "Block Coding with Noiseless Feedback": What is the maximum fraction r of errors from which a feedback-based error-correcting code can still uniquely be decoded? Remarkably, he proved that in the feedback setting, the unique decoding radius increases to $\frac{1}{3}$, surpassing the Plotkin bound.

[1] In this introduction, when we state a resilience threshold/radius r, we mean that for any $\epsilon > 0$ there exists a protocol resilient to $r - \epsilon$ errors, with rate $O_\epsilon(1)$. For sake of simplicity, we've omitted all ϵ's from this introduction.


© Meghal Gupta and Rachel Yun Zhang;
licensed under Creative Commons License CC-BY 4.0



While Berlekamp's result tells us the unique decoding radius of feedback codes, little is known about the list-decoding case. In list-decoding, first introduced by Elias [9] and Wozencraft [34], the goal is for Bob to output a small *list* of messages including Alice's. The concept of list decoding for non-feedback codes has proven to be quite illuminating for understanding the Hamming space and for various other applications (see e.g. [13] for further discussion).

In the classical setting, list decoding is useful because it can be done at a larger error radius than unique decoding. In fact, the optimal list decoding radius is known for (non-feedback) codes. For decoding into a list of size ℓ, the optimal radius is known to be $\frac{1}{2} - \binom{2s}{s} \cdot 2^{-2s-1}$ where $s := \lfloor \frac{\ell}{2} \rfloor$, which is asymptotically $\frac{1}{2} - \Theta(\ell^{-1/2})$ [5]. This is achieved by positive rate random codes.

In this paper, we investigate the analogous question for feedback codes, first posed in [30]. In a feedback code, what is the maximum fraction r of adversarial errors such that Bob can always learn a list of size ℓ containing Alice's message? We denote such a code as a (ℓ, r)-list decodable feedback code. In our work, we provide the first non-trivial unconditional lower and upper bounds for this question.

1.1 Results And Further Discussion

Our first result gives an upper bound on the error resilience (decoding radius) of an ℓ-list decodable feedback code.

▶ **Theorem 1.** *For any positive integer ℓ, there is no $\left(\ell, \frac{1}{2} - \frac{1}{2^{\ell+2}-2}\right)$-list feedback code.*

Prior to this work, the only known bound on the list decoding radius of feedback codes, aside from the trivial upper bound of $\frac{1}{2}$, was given in [30]. That result showed upper bounds for a specific subclass of feedback codes that satisfy a "generalized translation" property. It turns out this generalized translation property is not very comprehensive: we will in fact demonstrate a code that breaks the upper bound proven in [30]. In contrast, our upper bound is unconditional and applies to all feedback codes.

In the case of $\ell = 1$, Theorem 1 provides an upper bound of $\frac{1}{3}$ on the unique decoding radius, consistent with the result in [3]. The key question, however, is whether this bound is tight in general. Although we do not have an answer for arbitrary ℓ, we are able to show it is true in the specific case of $\ell = 2$. Specifically, we show that for any $\varepsilon > 0$, there exists a $\left(2, \frac{3}{7} - \varepsilon\right)$-list feedback code, matching the upper bound established by Theorem 1.

▶ **Theorem 2.** *For any $\varepsilon > 0$, there is a $\left(2, \frac{3}{7} - \varepsilon\right)$-list feedback code where $n = O_\varepsilon(k)$.*

The particular feedback code we used to achieve $\left(2, \frac{3}{7} - \epsilon\right)$-list decodability is the same code found in [31], which Spencer and Winkler showed achieves unique decoding up to radius $\frac{1}{3} - \epsilon$. The code, which we'll denote SW, is defined in Section 4. Compared to the $(\frac{1}{3} - \epsilon)$-unique decodability analysis in [31], however, our analysis is significantly more involved, requiring us to characterize the likelihood of four inputs rather than one possible input at once.

A reasonable conjecture at this point is that the SW protocol will always give a $\left(\ell, \frac{1}{2} - \frac{1}{2^{\ell+2}-2}\right)$-list decodable code. This would show that Theorem 1 is optimal.

Unfortunately, this turns out not to be true. We show that for $\ell = 3$, the error resilience of the Spencer-Winkler protocol does not match the upper bound given in Theorem 1. There is an adversarial corruption strategy on SW that requires only $\frac{31}{67} \approx 0.4627$ fraction of corruptions to confuse Bob between four different inputs, whereas the bound given by Theorem 1 is $\frac{7}{15} \approx 0.4667$.

▶ **Theorem 3.** *For any $\varepsilon > 0$, the feedback code* SW *(defined in Section 4) is not $\left(3, \frac{31}{67} + \varepsilon\right)$-list decodable for any choice of encoding length $n = n(k)$.*

Although this result states that the SW feedback code is not $\left(3, \frac{7}{15}\right)$-list decodable, it remains unknown whether a different feedback code achieves this bound or whether the upper bound can be improved.

Even the asymptotic behavior for this question remains unknown. Our upper bound suggests that the feedback list decoding radius could approach $\frac{1}{2}$ exponentially fast with respect to the list size ℓ. That is, the radius for ℓ-list decoding is given by $\frac{1}{2} - 2^{-O(\ell)}$. We conjecture that this exponential bound is in fact optimal for the feedback case and propose proving or disproving this as an interesting direction for future research.

In the case of standard ECC's (without feedback), as mentioned earlier, the optimal list decoding radius is $\frac{1}{2} - \binom{2s}{s} \cdot 2^{-2s-1}$ for $s = \left\lfloor \frac{\ell}{2} \right\rfloor$ [5]. That is, for any $\varepsilon > 0$, there exist $\left(\ell, \frac{1}{2} - \binom{2s}{s} \cdot 2^{-2s-1} - \varepsilon\right)$-list decodable codes. Asymptotically, this bound says that the list decoding radius is approximately $\frac{1}{2} - \Theta(\ell^{-1/2})$, which approaches $\frac{1}{2}$ much slower than the conjectured $\frac{1}{2} - 2^{-O(\ell)}$ radius for the feedback case. Even short of proving the exponential bound, demonstrating a separation between the asymptotics for the feedback and non-feedback cases would be an intriguing result.

1.2 Related Works

1.2.1 Error Correcting Codes with Feedback

We already mentioned the seminal work of [3] which introduced the concept of error-correcting codes with feedback. Berlekamp studied adversarial corruption strategies and proved that $\frac{1}{3}$ was the optimal error resilience for a feedback ECC in the face of adversarial errors. Feedback ECC's have also been studied for the binary symmetric channel, where every bit is flipped independently with some fixed probability [4, 35].

Particularly relevant is the work of [30], who initiated the study of list decoding feedback codes and gave some preliminary bounds on the list decoding capacities and radii of codes with noiseless feedback. Using generalizations of Berlekamp's methods, he provides upper bounds on list-decoding capacities for strategies that satisfy what he call a "generalized translation" property. By contrast, our upper bounds make no assumptions about the types of protocols allowed and are fully general.

Many other works on error-correcting codes with feedback present themselves instead under the lens of *adaptive search with lies*. The two setups are equivalent, and our results imply results in the field of adaptive search with lies. We discuss the relationship below.

1.2.1.1 Adaptive Search with Lies

An equivalent way of looking at error-correcting codes with unlimited feedback is as follows: After every message from Alice, Bob asks a yes or no question about Alice's input, and Alice's next bit is a response to that question. Then, the adversary can corrupt Alice's responses, but only up to some fraction α of the time. Bob is "searching" for Alice's input in a range $[N]$ by asking these yes/no questions, and receiving an incorrect answer up to an α fraction of the time. This game is known as the Rényi-Ulam game, independently introduced by Rényi in [27] and Ulam [33].

Two excellent surveys on the research in this area are presented in [8, 25]. There are a number of works that aspire to improve upon [3]. For example, [22] presents a strategy with optimal rate (asking the least possible number of questions) while still achieving the

best error-correction threshold. An alternative construction meeting the same optimal error resilience was given in [31]. We mention that some works focus on a fixed (constant) number of lies rather than resilience to a fraction of errors, and ask for the smallest number of questions necessary. [24] shows the optimal bound for one lie, the original Rényi-Ulam game. Followups find the optimal strategy for two and three lies as well [20, 7].

1.2.1.2 Limited Feedback/Questions

The works of [23, 1] study the minimal number of rounds, or *batches*, of questions Bob needs to ask when conducting adaptive search in the Ulam-Renyi model. The work of [25] poses the open question of how many batches of feedback are necessary before the optimal strategy becomes essentially the same as the fully adaptive setting.

In a more code-centric view, this model corresponds to a feedback code with *limited feedback*, where Alice receives feedback only a few times throughout the protocol instead of after every bit she sends. [21] studied the case where the feedback Alice receives consists of a small fraction of Alice's rounds. [12] took this direction even further, showing that a constant number of rounds of feedback consisting of only $\log(n)$ bits is enough to achieve the same unique decoding radius as for the unlimited feedback case. [29] showed that with $O(1)$ bits of feedback, one can achieve the zero-error capacity of the error channel with full noiseless feedback.

1.2.2 List Decoding

The notion of list decoding was introduced by Elias [9] and Wozencraft [34]. The goal is for Bob to output a small *list* of messages including Alice's. Although they introduced list-decoding in the setting of random error models, the main focus since then has been for adversarial error models. See [32, 16, 14] for surveys on the topic.

One main advantage is that list decoding enables correcting up to a factor two more worst-case errors compared to unique decoding. As we stated earlier, the optimal tradeoff between the list size ℓ and error tolerance (in the classical non-feedback setting) is known [5]. This work shows that there are positive rate codes achieving this maximum possible error tolerance. An analogous result for q-ary codes was shown in [28].

Another advantage of list-decoding is that it allows for better rate codes. One can ask what the optimal tradeoff between the error tolerance r and the rate R of the code is, when the list size is required to small or constant, but not necessarily achieving the optimal bounds above. In the large alphabet setting, a simple random coding argument shows that $r > 1 - R - o(1)$ is achievable. The work of [19] provides an explicit positive rate efficiently encodable/decodable code for this task using folded Reed Solomon codes. This research has since been expanded upon, including in [15, 6, 11, 2].

Works such as [18, 17] also discuss the tradeoff between all three parameters: list size, error tolerance, and rate.

2 Definitions

Before stating our results, let us formally define a list decodable feedback ECC.

▶ **Definition 4** ((ℓ, r)-List Decodable Feedback Code). *A (ℓ, r)-list feedback code is a family of protocols $\{\pi_k\}_{k\in\mathbb{N}}$ from Alice to Bob defined as follows for each $k \in \mathbb{N}$.*

- *Alice begins with an input $x \in \{0,1\}^k$ that she intends to communicate to Bob.*

- *In each of $n = n(k) = |\pi_k|$ rounds, Alice sends a single bit that may be flipped (by an adversary). At the end of the round, Alice learns whether the bit was flipped.*
- *At the end of the protocol, Bob must output a list L of size ℓ that must contain x, as long as at most rn bits were flipped throughout the protocol.*

The Coin Game Reformulation

Throughout the remainder of the paper, we'll argue our results about list decodable feedback ECC's in the language of a *coin game* played between two players Bob and Eve. To this end, we reformulate the task of communicating over a noisy channel with feedback as this aforementioned coin game. A version of this coin game reformulation was first given by [31] but is quite similar to the adaptive search with lies formulation of [33] and [27].

▶ **Definition 5** (($\ell, r; K, n$)-Coin Game). *In the* coin game *parametrized by $K, n, \ell \in \mathbb{N}$ and $r \in [0, 1]$, two players Bob and Eve are playing a game on a number line. At the beginning of the game, all K coins are at position 0. In each of n rounds, Bob partitions the coins into two sets, and Eve chooses a set for which Bob moves all the coins up by 1.*

Eve wins if, at the end of the n rounds, there are $\geqslant \ell + 1$ coins all with positions $\leqslant rn$, and Bob wins otherwise.

The connection to feedback codes is as follows. The $K = 2^k$ coins correspond to all possible values of Alice's input. For each of the n rounds of the feedback code, Bob partitions the coins into two sets based on the two possibilities 0 and 1 for Alice's next message. Based on the bit that he receives, he increments the position of all coins in the opposite set; that is, the coins for which Alice would have sent the opposite bit. In this manner, at any point in time, each coin's position is the number of corruptions that must have occurred so far if the given coin were actually Alice's input.

Let's say we want a scheme that is (ℓ, r)-list decodable. Bob wins if at the end of the n rounds, there are at most ℓ coins left with positions $< rn$ (we say all other coins have "fallen off the board"). Eve's goal, on the other hand, is to pick a set that Bob hears each round so that at the end of the game there are $> \ell$ coins left on the board. If she's able to pick *any* sequence of sets such that $> \ell$ coins are left on the board at the end of the game, then the resulting code is necessarily not (ℓ, r)-list decodable, since any of the $> \ell$ coins left on the board could've been Alice's input.

Notation

Let us define the notation that we will use for a coin game.

- For a coin x, $\mathsf{pos}_t(x)$ denotes the position of coin x after t rounds.
- $c_t(i)$ denotes the coin with the i'th smallest position at t rounds.
- For $i \in [K]$, $\mathsf{posc}_t(i) = \mathsf{pos}_t(c(i))$ denotes the position of the coin with the i'th smallest position after t rounds.

3 Upper Bound

In this section, we show an upper bound (impossibility result) that no ℓ-list feedback code can achieve an error resilience greater than $\frac{1}{2} - \frac{1}{2^{\ell+2}-2} = \frac{2^\ell - 1}{2^{\ell+1}-1}$. Let us state the main theorem of this section.

▶ **Theorem 1.** *For any positive integer ℓ, there is no $\left(\ell, \frac{1}{2} - \frac{1}{2^{\ell+2}-2}\right)$-list feedback code.*

In the formulation of the coin game, we wish to establish that in an n-round protocol with K coins, the adversary can always maintain $\ell+1$ coins below position $\frac{2^\ell-1}{2^{\ell+1}-1}\cdot n$ for any K sufficiently large compared to ℓ. From this, we formally conclude Theorem 1 in Section 3.3.

We begin by establishing the statement for $\ell = 1$; that is, the adversary can always maintain 2 coins at or below position $\frac{1}{3}\cdot n$. This statement was known as early as [3] when the concept of feedback codes was introduced, but we recreate a proof here. Our proof for general ℓ is inductive and therefore requires establishing the $\ell = 1$ case first.

3.1 Base Case: 1-List Feedback Codes

We want to show that the adversary can always maintain the second coin at or below position $\frac{1}{3}\cdot n$. However, in order to use our statement as an appropriate base case for the inductive hypothesis in Section 3.2, we actually show the following stronger statement about partially completed coin games.

Consider a partially completed coin game where m of the n rounds have been played already. Then, the adversary can limit $\mathsf{posc}_n(2)$ in terms of these variables, specifically by the expression $\max\left\{\left\lceil\frac{1}{3}\cdot(n-m+\mathsf{posc}_m(1)+\mathsf{posc}_m(2)+\mathsf{posc}_m(3))\right\rceil, \mathsf{pos}_m(2)\right\}$. In particular, the case where $m=0$ corresponds to the situation where the entire coin game is remaining, and indeed the expression says that $\mathsf{posc}_n(2)$ is at most $\frac{1}{3}\cdot n$.

▶ **Theorem 6.** *For any $\varepsilon > 0$, the following holds for sufficiently large n. Consider an n-round coin game on $K = 3$ coins in which m rounds have already been completed. Then, the adversary can ensure at the end of the n rounds that*

$$\mathsf{posc}_n(2) \leqslant \max\left\{\left\lceil\frac{1}{3}\cdot(n-m+\mathsf{posc}_m(1)+\mathsf{posc}_m(2)+\mathsf{posc}_m(3))\right\rceil, \mathsf{posc}_m(2)\right\}.$$

Before proving this, let us informally describe the strategy Eve uses to achieve this bound. This strategy is essentially identical to what has appeared in previous literature such as [10]. For most of the protocol, she simply chooses the smaller (fewer elements) of the two sets given by Bob. That way, she need only move up one coin, and so on average, a coin move up by $\frac{1}{3}$, which is what gives the $\frac{1}{3}\cdot(n-m)$ term in the stated bound. However, the issue is that the coins need not move evenly. For example, the first coin may not move at all, while the last two coins each move by $\frac{1}{2}\cdot(n-m)$. Then, the position of the second coin increases too much, but in compensation, we have some leeway with the first coin. To use this, once the first coin is sufficiently low in position that it can never surpass the second coin, Eve switches strategies to always choosing the set that *does not* contain the second coin. Now, although the average coin can move up more per round, the second coin no longer moves which is what we wanted, and we do not risk the first coin catching up to the second by doing this.

Proof. Let $\theta = \max\left\{\left\lceil\frac{1}{3}\cdot(n-m+\mathsf{posc}_m(1)+\mathsf{posc}_m(2)+\mathsf{posc}_m(3))\right\rceil, \mathsf{posc}_m(2)\right\}$. In each round t, Bob chooses the set S_t and its complement $\overline{S_t}$ (both subsets of the 3 coins in the game), and the adversary Eve needs to choose one set for which to move up the coins by 1 each. Eve uses the following strategy.

For a round t, while $\mathsf{posc}_{t-1}(2) < \theta$, Eve always chooses the smaller of S_t and $\overline{S_t}$, that is the one with fewer elements. (Note that the inequality in this condition is strict, so, for example, if $\mathsf{posc}_m(2) \geqslant \left\lceil\frac{1}{3}\cdot(n-m+\mathsf{posc}_m(1)+\mathsf{posc}_m(2)+\mathsf{posc}_m(3))\right\rceil$, it is possible to skip this step entirely.)

Once $\mathsf{posc}_t(2)$ reaches θ, then Eve instead always selects the set not containing the coin in the second position.

Let us say this transition between the two strategies occurs at round $T+1$, so Eve spends rounds $m+1$ through T inclusive doing the first strategy, and the rest of the protocol doing the second strategy. (If this round T never occurs, then at the end of the protocol, $\mathsf{posc}_n(2) < \theta$ as we desired.) Our goal will be to show that in the remaining $n-T$ rounds, the first coin will not pass the second coin, even if it is selected to increase every time. Then there will be two coins left at the end of the protocol, both with position $\leqslant \theta$.

Between rounds $m+1$ and T, in every step, the total positions of the coins are increasing by at most 1. Also, the first time the inequality is violated, we must have $\mathsf{posc}_T(2) = \theta \geqslant \left\lceil \frac{1}{3} \cdot (n - m + \mathsf{posc}_m(1) + \mathsf{posc}_m(2) + \mathsf{posc}_m(3)) \right\rceil$. Therefore, at the end of round T, it holds that

$$\begin{aligned}
&\mathsf{posc}_T(1) + \mathsf{posc}_T(2) + \mathsf{posc}_T(3) \leqslant \mathsf{posc}_m(1) + \mathsf{posc}_m(2) + \mathsf{posc}_m(3) + T - m \\
\implies &\mathsf{posc}_T(1) + 2\mathsf{posc}_T(2) \leqslant \mathsf{posc}_m(1) + \mathsf{posc}_m(2) + \mathsf{posc}_m(3) + T - m \\
\implies &\mathsf{posc}_T(1) - \mathsf{posc}_T(2) + 3 \cdot \left\lceil \frac{1}{3} \cdot (n - m + \mathsf{posc}_m(1) + \mathsf{posc}_m(2) + \mathsf{posc}_m(3)) \right\rceil \\
&\qquad \leqslant \mathsf{posc}_m(1) + \mathsf{posc}_m(2) + \mathsf{posc}_m(3) + T - m \\
\implies &\mathsf{posc}_T(1) - \mathsf{posc}_T(2) \\
&\qquad \leqslant \mathsf{posc}_m(1) - \mathsf{posc}_m(2) + \mathsf{posc}_m(3) + T - m - (n - m + \mathsf{posc}_m(1) + \mathsf{posc}_m(2) + \mathsf{posc}_m(3)) \\
&\qquad \leqslant T - n \\
\implies &\mathsf{posc}_T(2) - \mathsf{posc}_T(1) \geqslant n - T
\end{aligned}$$

In other words, in the remaining $n-T$ rounds, the first coin will not be able to switch positions with the second or third coin. As such, $\mathsf{posc}(2)$ will never increase since it is never selected, and the first coin never surpasses it.

This concludes the proof, as it means that $\mathsf{posc}_n(2)$ stays at θ (or never reached θ). ◀

3.2 Inductive Step: ℓ-List Feedback Codes

In this section, we establish the inductive step. We inductively show an analog of Theorem 6 for larger list sizes.

▶ **Theorem 7.** *For any positive integer ℓ, the following holds for sufficiently large n. Consider an n-round coin game on $K = 2^{\ell+1} - 1$ coins in which m rounds have already been completed. For round $m \in [n]$, let*

$$\theta_\ell := \left\lceil \frac{1}{2^{\ell+1} - 1} \cdot \left((2^\ell - 1) \cdot (n - m) + \mathsf{posc}_m(1) + \ldots + \mathsf{posc}_m(K) \right) \right\rceil.$$

If $\mathsf{posc}_m(1) \geqslant \theta_\ell + m - n$ then Eve can ensure at the end of the n rounds that $\mathsf{posc}_n(\ell+1) \leqslant \theta_\ell$.

Again, before proving this formally, let us describe Eve's strategy at a high level. Her strategy to achieve this bound will be recursive. At the start of the protocol, she chooses the smaller of Bob's two sets at every step, so that the average coin moves by $\frac{2^\ell - 1}{2^{\ell+1} - 1}$, giving the $\frac{2^\ell - 1}{2^{\ell+1} - 1} \cdot (n - m)$ term in the bound. As with the $\ell = 1$ case, this doesn't give the desired bound for the final position of the $(\ell+1)$'th coin because it's possible the first few coins did not move at all, while the $(\ell+1)$'th coin moved by more. However, it can be shown that in compensation, the position of the first coin must be very low in this case. Once the first coin is at a low enough position that even if it is selected in every remaining round, it can no longer exceed θ_ℓ by the end of the protocol, Eve switches strategies. Now, she needs to ensure that of the remaining coins, ℓ of them remain below position θ_ℓ (since we have already found one coin that will definitely remain below θ_ℓ). This is where her strategy becomes recursive: she simulates a new coin game on the leftmost $2^\ell - 1$ coins of the $2^{\ell+1} - 2$ remaining coins and plays on these coins according to her (recursively described) strategy for the case of $\ell - 1$.

Proof. First we check that the statement holds for the base case $\ell = 1$. This follows from Theorem 6. In this case, we have that

$$\begin{aligned}\theta_1 &:= \left\lceil \frac{1}{3} \cdot (n - m + \mathsf{posc}_m(1) + \mathsf{posc}_m(2) + \mathsf{posc}_m(3)) \right\rceil \\ &\geqslant \frac{1}{3} \cdot (n - m + \mathsf{posc}_m(1) + \mathsf{posc}_m(2) + \mathsf{posc}_m(3)) \\ &\geqslant \frac{1}{3} \cdot (\theta_1 + 2\mathsf{posc}_m(2)).\end{aligned}$$

In the last step, we used the condition in the Theorem 7 that $\mathsf{posc}_m(1) \geqslant \theta_\ell + m - n$. Rearranging, this implies that $\theta_1 \geqslant \mathsf{posc}_m(2)$. Theorem 6 states that $\mathsf{posc}_n(2) \leqslant \max\{\theta_1, \mathsf{posc}_m(2)\} = \theta_1$, since $\theta_1 \geqslant \mathsf{posc}_m(2)$, which is what we wanted to prove.

For the inductive step, let us assume the statement is true for $\ell - 1$ and prove it for ℓ. Throughout the proof, let $\Sigma_m := \mathsf{posc}_m(1) + \ldots + \mathsf{posc}_m(K)$.

In each round t, Bob's two sets are S_t and its complement $\overline{S_t}$, and the adversary Eve needs to choose one set for which to move up the coins by 1 each. Eve uses the following strategy.

For a round t, while $\mathsf{posc}_{t-1}(1) > \theta_\ell + (t-1) - n$, Eve always chooses the one of S_t and $\overline{S_t}$ with fewer elements. Note that the first time it is violated, $\mathsf{posc}_t(1) = \theta_\ell + t - n$, even if it is already violated immediately at round $t = m + 1$ by the assumption. Denote the round it is first violated as T if it is ever violated.

At this point (after round T) or at the start of the protocol if the condition began by being violated, it is impossible for the coin $c_T(1)$ to ever hit the threshold θ_ℓ, since its position can only increase by 1 for each of the remaining $n - T$ rounds. Thus Eve can ignore the first coin, and play a coin game on only the remaining $2^{\ell+1} - 1$ coins, with the goal of keeping $\mathsf{posc}_n(\ell)$ below θ_ℓ. She also opts to exclude the second half of these coins $c_T(2^\ell + 1), \ldots, c_T(2^{\ell+1} - 1)$ from her new coin game as well, and simply assumes that they will end up above the threshold. In all, the players are playing a new a coin game on $2^\ell - 1$ coins which correspond to $c_T(2), \ldots, c_T(2^\ell)$, with Eve having the goal of keeping $\mathsf{posc}_n(\ell)$ below θ_ℓ. She executes the strategy for this recursively, keeping $\mathsf{posc}_n(\ell)$ bounded by what Theorem 7 says for $\ell - 1$.

Now, let us analyze this strategy.

First, we address the case where round T never occurs. In this case, at the end of the protocol, it holds that $\mathsf{posc}_n(1) > \theta_\ell - n + n = \theta_\ell$. Since Eve is picking the smaller of S_t and $\overline{S_t}$ each time, it holds that

$$\begin{aligned}&\mathsf{posc}_n(1) + \ldots + \mathsf{posc}_n(2^{\ell+1} - 1) \leqslant (2^\ell - 1) \cdot (n - m) + \Sigma_m \\ \implies &\mathsf{posc}_n(1) \leqslant \frac{1}{2^{\ell+1} - 1} \cdot \left((2^\ell - 1) \cdot (n - m) + \Sigma_m\right) \leqslant \theta_\ell,\end{aligned}$$

which is a contradiction, where the last inequality follows because the middle term is at most θ_ℓ by definition.

Next, we address the case where round T occurs at some time $m \leqslant T \leqslant n$. After round T, it holds that $\mathsf{posc}_T(1) = \theta_\ell + T - n$ exactly, and so

$$\begin{aligned}&\mathsf{posc}_T(1) + \ldots + \mathsf{posc}_T(2^{\ell+1} - 1) \leqslant (2^\ell - 1) \cdot (T - m) + \Sigma_m \\ \implies &\mathsf{posc}_T(2) + \ldots + \mathsf{posc}_T(2^{\ell+1} - 1) \leqslant (2^\ell - 2) \cdot T - (2^\ell - 1) \cdot m - \theta_\ell + n + \Sigma_m \\ \implies &\mathsf{posc}_T(2) + \ldots + \mathsf{posc}_T(2^\ell) \leqslant (2^{\ell-1} - 1) \cdot T - \frac{2^\ell - 1}{2} \cdot m + \frac{1}{2} \cdot (n + \Sigma_m - \theta_\ell). \qquad (1)\end{aligned}$$

The coin in position 1 at time T can never fall off the board, because its position can only increment by 1 in each of the last $T-n$ rounds, and so we can disregard it. We will now apply the inductive hypothesis to a new game played on the $2^\ell - 1$ coins in positions 2 through 2^ℓ at time T, corresponding to the coins in positions 2 through 2^ℓ at time T. We will renumber the coins, so we can view this new game as being played on $2^\ell - 1$ coins numbered 1 through $2^\ell - 1$, where the position functions are given by pos' and posc'. Denote

$$\theta'_{\ell-1} := \left\lceil \frac{1}{2^\ell - 1} \cdot \left((2^{\ell-1} - 1) \cdot (n - T) + \mathsf{posc}'_T(1) + \ldots + \mathsf{posc}'_T(2^\ell - 1) \right) \right\rceil.$$

Our goal is to show that $\theta'_{\ell-1} \leqslant \theta_\ell$. This will prove the inductive step for the following reasons:

- We have that $\mathsf{posc}'_T(1) = \mathsf{posc}_T(2) \geqslant \mathsf{posc}_T(1) = \theta_\ell - T + n \geqslant \theta'_{\ell-1} - T + n$, so the condition for the inductive step of the theorem is met.
- We know that $\mathsf{posc}_n(\ell+1) \leqslant \max\{\mathsf{posc}'_n(\ell), \theta_\ell\}$, because coin $c_T(1)$ cannot surpass θ_ℓ and all of coins $c_T(2) \ldots c_T(\ell+1)$ are at most $\mathsf{posc}'_n(\ell)$. Moreover, by the inductive hypothesis, we know that $\mathsf{posc}'_n(\ell) < \theta'_{\ell-1} \leqslant \theta_\ell$, and so $\mathsf{posc}_n(\ell+1) \leqslant \theta_\ell$ as we wanted.

Now, we show the desired statement that $\theta'_{\ell-1} \leqslant \theta_\ell$. Using the bound (1) we found on $\mathsf{posc}_T(2) + \ldots + \mathsf{posc}_T(2^\ell) = \mathsf{posc}'_T(1) + \ldots + \mathsf{posc}'_T(2^\ell - 1)$, we have that

$$\begin{aligned}
\theta'_{\ell-1} &= \left\lceil \frac{1}{2^\ell - 1} \cdot \left((2^{\ell-1} - 1) \cdot (n - T) + \mathsf{posc}'_T(1) + \ldots + \mathsf{posc}'_T(2^\ell - 1) \right) \right\rceil \\
&< \left\lceil \frac{1}{2^\ell - 1} \cdot \left((2^{\ell-1} - 1) \cdot (n - T) + (2^{\ell-1} - 1) \cdot T - \frac{2^\ell - 1}{2} \cdot m + \frac{1}{2} \cdot (n + \Sigma_m - \theta_\ell) \right) \right\rceil \\
&= \left\lceil \frac{1}{2^{\ell+1} - 2} \cdot \left((2^\ell - 1) \cdot (n - m) + \Sigma_m - \theta_\ell \right) \right\rceil \\
&\leqslant \left\lceil \frac{1}{2^{\ell+1} - 2} \cdot \left(1 - \frac{1}{2^{\ell+1} - 1} \right) \left((2^\ell - 1) \cdot (n - m) + \Sigma_m \right) \right\rceil \\
&= \left\lceil \frac{1}{2^{\ell+1} - 1} \cdot \left((2^\ell - 1) \cdot (n - m) + \Sigma_m \right) \right\rceil = \theta_\ell.
\end{aligned}$$

This concludes the inductive step of Theorem 7. ◀

3.3 Conclude Theorem 1

Finally, let us conclude Theorem 1. Let the number of coins K be at least $8 \cdot (2^{\ell+1} - 1)$. For Eve's first move, have her choose the one of S_1 and $\overline{S_1}$ (Bob's two provided sets) with fewer elements, so that at least half the coins remain at position 0. For her second move, have her choose the one of S_2 and $\overline{S_2}$ that keeps the most coins at 0, which will keep at leat $\frac{1}{4}$ of the coins at 0. She will do this one more time, so that after 3 moves, there are at least $2^{\ell+1} - 1$ coins at position 0.

Even if there are more than $2^{\ell+1} - 1$ coins at position 0, from now, Eve will fix a subset of $2^{\ell+1} - 1$ coins to play a coin game on. She wants to show that in this new $n-3$ turn coin game, she can keep $\mathsf{posc}_{n-3}(\ell+1) \leqslant \frac{2^\ell - 1}{2^{\ell+1} - 1} \cdot n$. This follows by setting $m = 0$ in Theorem 7. The theorem states that Eve can always guarantee that $\ell+1$ coins stay at or below position $\left\lceil \frac{2^\ell - 1}{2^{\ell+1} - 1} \cdot (n-3) \right\rceil \leqslant \frac{2^\ell - 1}{2^{\ell+1} - 1} \cdot n$. Therefore $\left(\ell+1, \frac{2^\ell - 1}{2^{\ell+1} - 1}\right)$-list decoding is not possible.

4 Optimal 2-List Feedback Codes

In this section, we will prove that the feedback ECC from Spencer-Winkler [31] achieves $\left(2, \frac{3}{7} - \epsilon\right)$-list decodability, matching the upper bound for $\ell = 2$ given in Theorem 1.

4.1 The Spencer-Winkler Coin Game

Let us begin by defining the Spencer-Winkler strategy for the coin game.

Protocol 1 : The Spencer-Winkler (SW) Coin Game

Alice has a coin $x \in \{0,1\}^k$. Bob and Eve play the coin game with $K = 2^k$ coins. In each of n rounds, Bob partitions the coins into two sets based on the parity of their positions on the board. That is, he numbers the coins from 1 to K, where 1 is assigned to the coin with the smallest position, and K is the coin that is furthest away. The odd numbered coins are in one set, and the even numbered coins are in the other. Eve chooses a set whose coins to move up by 1.

We will also denote by SW the feedback code corresponding to this family of coin game strategies. When defining a feedback code SW, we must also specify a choice of the code length $n(k)$. It was shown in [31] that SW achieves the unique decoding bound of $\frac{1}{3} - \epsilon$ for feedback codes.

▶ **Theorem 8** ([31]). *Let $n = \frac{k}{\epsilon}$. Then in the* SW *coin game, no matter which sets Eve chooses each round, it will hold that*

$$\mathsf{posc}_n(2) > \left(\frac{1}{3} - O(\epsilon)\right) \cdot n.$$

Thus the SW *coin game gives a feedback code that is uniquely decodable up to radius $\frac{1}{3} - O(\epsilon)$.*

Our goal through the rest of this section is to prove a statement about the list decodability of SW to list size 2. This proves Theorem 2.

▶ **Theorem 9.** *In the* SW *coin game, no matter which sets Eve chooses each round, it holds that*

$$\mathsf{posc}_n(3) > \left(\frac{3}{7} - 2\epsilon\right) \cdot n.$$

That is, the SW *coin game with $n = \frac{24k}{\epsilon^3}$ gives a feedback code that is $\left(2, \frac{2}{7} - 2\epsilon\right)$-list-decodable.*

4.2 Some Useful Inequalities

Throughout this section, let $n = \frac{24k}{\epsilon^3}$. Our proof that the SW protocol achieves $(2, \frac{3}{7})$-list decoding follows from two inequalities, given in Lemmas 11 and 12. First, let us prove a statement about how the position of the i'th coin at each round can change.

▷ Claim 10. For any $t \in [n-1]$ and for any $i \in [K]$, it holds that

$$\mathsf{posc}_{t+1}(i) - \mathsf{posc}_t(i) \in \{0, 1\}.$$

Proof. The i'th coin $c_{t+1}(i)$ at time $t+1$ being at position $\mathsf{posc}_{t+1}(i)$ means that at time t there must've been $\geqslant i-1$ coins with positions $\leqslant \mathsf{posc}_{t+1}(i)$ and $\geqslant K - i$ coins with positions $\geqslant \mathsf{pos}_t(c_{t+1}(i)) \geqslant \mathsf{posc}_{t+1}(i) - 1$. This means that the i'th coin at time t must've had position in the range $[\mathsf{posc}_{t+1}(i) - 1, \mathsf{posc}_{t+1}(i)]$, from which the claim follows. ◁

Now, we are ready to state our first key lemma, that the gaps between the positions of the first two coins must always be at least the gap between the positions of the third and fourth coins at any timestep.

▶ **Lemma 11.** *At any point in the* SW *protocol, the following inequality holds:*

$$\mathsf{posc}_t(2) - \mathsf{posc}_t(1) + 1 \geqslant \mathsf{posc}_t(4) - \mathsf{posc}_t(3). \tag{2}$$

Proof. We prove this via straightforward induction. Let us assume at time t that $\mathsf{posc}_t(2) - \mathsf{posc}_t(1) + 1 \geqslant \mathsf{posc}_t(4) - \mathsf{posc}_t(3)$. This is certainly true at time $t = 0$, since all coins are at position 0.

By applying Claim 10, it follows that $\mathsf{posc}_{t+1}(2) - \mathsf{posc}_{t+1}(1) = \mathsf{posc}_t(2) - \mathsf{posc}_t(1) + \{-1, 0, 1\}$ and similarly $\mathsf{posc}_{t+1}(4) - \mathsf{posc}_{t+1}(3) = \mathsf{posc}_t(4) - \mathsf{posc}_t(3) + \{-1, 0, 1\}$. This means that $(\mathsf{posc}(4) - \mathsf{posc}(3)) - (\mathsf{posc}(2) - \mathsf{posc}(1))$ can change by at most 2 each round, and we can focus on the relevant case that

$$(\mathsf{posc}_t(4) - \mathsf{posc}_t(3)) - (\mathsf{posc}_t(2) - \mathsf{posc}_t(t)) \in \{0, 1\}. \tag{3}$$

We split the induction into two cases. The first is if Bob moves the odd set. In this case, Bob will move up coins $c_t(1)$ and $c_t(3)$. If $c_t(4) - c_t(3) > 0$, then this means that $c_t(4) - c_t(3)$ will decrease by 1, and $c_t(2) - c_t(1)$ will change by at most 1, so (2) will still hold at time $t + 1$. If $c_t(4) - c_t(3) = 0$, then $c_{t+1}(4) - c_{t+1}(3) \leqslant 1$, and also by (3) it must be true that $\mathsf{posc}_t(2) - \mathsf{posc}_t(1) = 0$, so $c_{t+1}(2) - c_{t+1}(1) \in \{0, 1\}$ as well. Therefore, $(c_{t+1}(4) - c_{t+1}(3)) - (c_{t+1}(2) - c_{t+1}(1)) \leqslant 1$ as desired.

The second case is if Bob moves the even set. In this case, Bob will move up coins $c_t(2)$ and $c_t(4)$. If $\mathsf{posc}_t(3) - \mathsf{posc}_t(2) > 0$, then $c_t(1), c_t(2)$, and $c_t(3)$ will still be the first, second, and third coins at time $t+1$. Then $\mathsf{posc}_{t+1}(2) - \mathsf{posc}_{t+1}(1) = (\mathsf{posc}_t(2) + 1) - \mathsf{posc}_t(1)$, while $\mathsf{posc}_{t+1}(4) - \mathsf{posc}_{t+1}(3) \leqslant (\mathsf{posc}_t(4) + 1) - \mathsf{posc}_t(3)$ by Claim 10, so (2) clearly follows. On the other hand, if $\mathsf{posc}_t(2) = \mathsf{posc}_t(3)$, then moving coins the second and fourth coins at time t is equivalent to moving the third and fourth coins at time t, so $\mathsf{posc}_{t+1}(2) - \mathsf{posc}_{t+1}(1) = \mathsf{posc}_t(2) - \mathsf{posc}_t(1)$. If $\mathsf{posc}_t(4) > \mathsf{posc}_t(3)$, then $\mathsf{posc}_{t+1}(3) = \mathsf{posc}_t(3) + 1$, so $\mathsf{posc}_{t+1}(4) - \mathsf{posc}_{t+1}(3) \leqslant \mathsf{posc}_t(4) - \mathsf{posc}_t(3)$, and (2) follows. Otherwise if $\mathsf{posc}_t(4) - \mathsf{posc}_t(3) = 0$, this means by (3) that $\mathsf{posc}_t(2) - \mathsf{posc}_t(1) = 0$, and since $\mathsf{posc}_{t+1}(4) = \mathsf{posc}_t(4) + \{0, 1\}$ and $\mathsf{posc}_{t+1}(3) = \mathsf{posc}_t(3) + \{0, 1\}$ by Claim 10, we have that $\mathsf{posc}_{t+1}(4) - \mathsf{posc}_{t+1}(3) \leqslant 1$, so again (2) follows. ◀

The second key lemma is about the sum of the positions of the first four coins at any timestep.

▶ **Lemma 12.** *At any round $0 \leqslant \tau \leqslant n$, the following holds:*

$$\frac{1}{2}\mathsf{posc}_\tau(1) + \mathsf{posc}_\tau(2) + \mathsf{posc}_\tau(3) + \mathsf{posc}_\tau(4) \geqslant \frac{3\tau}{2} - 5\epsilon n.$$

Proof. In order to analyze the positions of four coins collectively, we define a new derivative coin game that we call the *quadruple coin game*. The coins in this quadruple coin game are four-tuples of coins in the old game, for a total of $\binom{K}{4}$ coins in the new game. Let the set of coins in the original game be denoted S, and let the set of quadruple coins be denoted S^{q}. The position of quadruple coin $(x_1, x_2, x_3, x_4) \in S^{\mathsf{q}}$ at time t is $\frac{1}{2}\mathsf{pos}_t(x_1) + \mathsf{pos}_t(x_2) + \mathsf{pos}_t(x_3) + \mathsf{pos}_t(x_4)$, where here we assume that the four coins x_1, x_2, x_3, x_4 are ordered according to increasing position at time t.

We denote by $c_t^{\mathsf{q}}(i_1, i_2, i_3, i_4)$ the quadruple coin $(c_t(i_1), c_t(i_2), c_t(i_3), c_t(i_4))$, and by $\mathsf{posc}_t^{\mathsf{q}}(i_1, i_2, i_3, i_4) = pos_t^{\mathsf{q}}(c_t(i_1, i_2, i_3, i_4))$ its position at time t. Our goal is to show that in this quaduple coin game, coin $c_n(1, 2, 3, 4)$ will be at position $\frac{3}{2}n - 4\epsilon n$ at the end of the protocol.

To do this, we will show that on average, the quadruple coins must be moving at $\approx \frac{3}{2}$ units per round, and furthermore that no coins are left behind. What we mean by the latter is that for any quadruple coin that doesn't move enough in a given round, there is some earlier coin that is moving more than enough and compensating for the slower movement by the first coin. Precisely, we will do this via a bijection.

Let us first define $u_t(i)$ to be the amount that coin $c_t(i)$ is moved in round $t+1$. We remark that this is *not* the value $\mathsf{posc}_{t+1}(i) - \mathsf{posc}_t(i)$, since after round t, the coins may change their orders so that $c_t(i)$ is no longer the i'th coin. We will also define $u_t^{\mathfrak{q}}(i_1, i_2, i_3, i_4)$ to be the amount that quadruple coin $c_t^{\mathfrak{q}}(i_1, i_2, i_3, i_4)$ is moved in time $t+1$. In general, when we write a 4-tuple, the four numbers should be in increasing order. We can also define the poset relation on 4-tuples of indices in $[K]$ by $(i_1, i_2, i_3, i_4) < (j_1, j_2, j_3, j_4)$ if $i_\iota \leqslant j_\iota$ for all $\iota \in [4]$.

▷ Claim 13. At any time t there is a bijection of indices $\binom{[K]}{4} \ni (i_1, i_2, i_3, i_4) \leftrightarrow (j_1, j_2, j_3, j_4) \in \binom{[K]}{4}$ such that for any bijected values $(i_1, i_2, i_3, i_4) \leftrightarrow (j_1, j_2, j_3, j_4)$,

$$u_t^{\mathfrak{q}}(i_1, i_2, i_3, i_4) + u_t^{\mathfrak{q}}(j_1, j_2, j_3, j_4) \geqslant 3,$$

and furthermore if $u_t^{\mathfrak{q}}(i_1, i_2, i_3, i_4) < \frac{3}{2}$ then $(j_1, j_2, j_3, j_4) < (i_1, i_2, i_3, i_4)$.

Proof. We will define this bijection φ_t by defining φ_t on tuples (i_1, i_2, i_3, i_4) for which $u_t^{\mathfrak{q}}(i_1, i_2, i_3, i_4) < \frac{3}{2}$. For all other tuples that remain unbijected after this process, we can pair them up arbitrarily.

Consider a tuple (i_1, i_2, i_3, i_4) for which $u_t^{\mathfrak{q}}(i_1, i_2, i_3, i_4) < \frac{3}{2}$. Let's suppose that Bob is moving the even coins in round $t+1$ (we'll define the bijection for the odd case in the same way, except switching the words "even" and "odd" everywhere). The only way $u_t^{\mathfrak{q}}(i_1, i_2, i_3, i_4)$ can be less than $\frac{3}{2}$ is if there is at most one even index among the four. Let us define the map φ_t depending on which of the i_ι is even.

$$\varphi_t(i_1, i_2, i_3, i_4) = \begin{cases} (i_1, i_2 - 1, i_3 - 1, i_4 - 1) & i_1, i_2, i_3, i_4 \text{ all odd} \\ (i_1, i_2, i_3 - 1, i_4) & i_1, i_2, i_3 \text{ odd and } i_4 \text{ even} \\ (i_1, i_2 - 1, i_3, i_4) & i_1, i_2, i_4 \text{ odd and } i_3 \text{ even} \\ (i_1, i_2, i_3, i_4 - 1) & i_1, i_3, i_4 \text{ odd and } i_2 \text{ even} \\ (i_1, i_2, i_3, i_4 - 1) & i_2, i_3, i_4 \text{ odd and } i_1 \text{ even.} \end{cases}$$

It's straightforward to check that these are all well defined, and that each of the bijected tuples contains four distinct elements. (Recall we always assume $i_1 < i_2 < i_3 < i_4$.)

Next, we also need to check that $\varphi_t(i_1, i_2, i_3, i_4) \neq \varphi_t(i'_1, i'_2, i'_3, i'_4)$ for all $(i_1, i_2, i_3, i_4) \neq (i'_1, i'_2, i'_3, i'_4)$. To see this, we show how to invert each tuple in the image of the map φ_t as we've defined so far. The key observation is that the five cases give five different parity tuples for the output tuple: for instance, the first case that i_1, i_2, i_3, i_4 are all odd results in an output where the parities are (odd, even, even, even), whereas all the rest of the maps give outputs where two indices are odd and two are even, so we always know how to invert a tuple that is (odd, even, even, even).

$$\varphi_t^{-1}(j_1, j_2, j_3, j_4) = \begin{cases} (j_1, j_2 + 1, j_3 + 1, j_4 + 1) & j_1 \text{ odd and } j_2, j_3, j_4 \text{ odd} \\ (j_1, j_2, j_3 + 1, j_4) & j_1, j_2 \text{ odd and } j_3, j_4 \text{ even} \\ (j_1, j_2 + 1, j_3, j_4) & j_1, j_4 \text{ odd and } j_2, j_3 \text{ even} \\ (j_1, j_2, j_3, j_4 + 1) & j_1, j_3 \text{ odd and } j_2, j_4 \text{ even} \\ (j_1, j_2, j_3, j_4 + 1) & j_2, j_3 \text{ odd and } j_1, j_4 \text{ even.} \end{cases}$$

Finally, we check that $u_t^{\mathsf{q}}(i_1,i_2,i_3,i_4) + u_t^{\mathsf{q}}(\varphi_t(i_1,i_2,i_3,i_4)) \geqslant 3$. In the first case that i_1,i_2,i_3,i_4 are all odd and $_t^{\mathsf{q}}(i_1,i_2,i_3,i_4) = 0$, $\varphi_t(i_1,i_2,i_3,i_4)$ has three even indices, so $u_t^{\mathsf{q}}(\varphi_t(i_1,i_2,i_3,i_4)) = 3$. In all the other cases, (i_1,i_2,i_3,i_4) has one even index and $\varphi_t(i_1,i_2,i_3,i_4)$ has two even indices so $u_t^{\mathsf{q}}(i_1,i_2,i_3,i_4) + u_t^{\mathsf{q}}(\varphi_t(i_1,i_2,i_3,i_4)) = 1+2 = 3$. ◁

Now, let us define the potential function

$$\psi(t) = \sum_{(x_1,x_2,x_3,x_4)\in\binom{\{0,1\}^k}{4}} w_t(x_1,x_2,x_3,x_4),$$

where

$$w_t(x_1,x_2,x_3,x_4) = (1+\epsilon)^{\frac{1}{3}\cdot d_t(x_1,x_2,x_3,x_4)}$$

and

$$d_t(x_1,x_2,x_3,x_4) = \max\left\{0, \frac{3t}{2} - \mathsf{pos}_t^{\mathsf{q}}(x_1,x_2,x_3,x_4)\right\}.$$

Since $\mathsf{pos}_0^{\mathsf{q}}(x_1,x_2,x_3,x_4) = 0$ for all (x_1,x_2,x_3,x_4) at the start of the game, $\psi(0) = \binom{K}{4}$.

▷ Claim 14. For any time t, let us consider any bijected pair $(i_1,i_2,i_3,i_4) \leftrightarrow (j_1,j_2,j_3,j_4)$ under φ_t. Define $(x_1,x_2,x_3,x_4) = c_t^{\mathsf{q}}(i_1,i_2,i_3,i_4)$ and $(y_1,y_2,y_3,y_4) = c_t^{\mathsf{q}}(j_1,j_2,j_3,j_4)$. Then,

$$w_{t+1}(x_1,x_2,x_3,x_4) + w_{t+1}(y_1,y_2,y_3,y_4) \leqslant e^{\epsilon^2/2}\cdot(w_t(x_1,x_2,x_3,x_4) + w_t(y_1,y_2,y_3,y_4)) + \epsilon,$$

Proof. We proceed by casework on whether $u_t(x_1,x_2,x_3,x_4)$ and $u_t(y_1,y_2,y_3,y_4)$ are greater than or less than $\frac{3}{2}$.

Case 1: *Both $u_t(x_1,x_2,x_3,x_4)$ and $u_t(y_1,y_2,y_3,y_4)$ are at least $\frac{3}{2}$.*

In this case, $d_{t+1}(x_1,x_2,x_3,x_4) \leqslant d_t(x_1,x_2,x_3,x_4)$ and $d_{t+1}(y_1,y_2,y_3,y_4) \leqslant d_t(y_1,y_2,y_3,y_4)$, so

$$\begin{aligned}
&w_{t+1}(x_1,x_2,x_3,x_4) + w_{t+1}(y_1,y_2,y_3,y_4)\\
&\quad\leqslant w_t(x_1,x_2,x_3,x_4) + w_t(y_1,y_2,y_3,y_4)\\
&\quad\leqslant e^{\epsilon^2/2}\cdot(w_t(x_1,x_2,x_3,x_4) + w_t(y_1,y_2,y_3,y_4)) + \epsilon.
\end{aligned}$$

Case 2: $u_t(x_1,x_2,x_3,x_4) < \frac{3}{2}$.

We write $u_t(x_1,x_2,x_3,x_4) = \frac{3}{2} - \delta$ for some $\delta > 0$. Then $d_{t+1}(x_1,x_2,x_3,x_4) \leqslant d_t(x_1,x_2,x_3,x_4) + \delta$ and $u_t(y_1,y_2,y_3,y_4) \geqslant \frac{3}{2} - \delta$. Let us again do casework on whether $d_t(y_1,y_2,y_3,y_4)$ is larger or smaller than δ.

Subcase 2.1: $d_t(y_1,y_2,y_3,y_4) \geqslant \delta$.

In this case, $d_{t+1}(y_1,y_2,y_3,y_4) \leqslant d_t(y_1,y_2,y_3,y_4) - \delta$. Then

$$\begin{aligned}
&w_{t+1}(x_1,x_2,x_3,x_4) + w_{t+1}(y_1,y_2,y_3,y_4)\\
&\quad\leqslant (1+\epsilon)^{\frac{1}{3}\delta} w_t(x_1,x_2,x_3,x_4) + (1+\epsilon)^{-\frac{1}{3}\delta} w_t(y_1,y_2,y_3,y_4)\\
&\quad\leqslant \frac{(1+\epsilon)^{\frac{1}{3}\delta} + (1+\epsilon)^{-\frac{1}{3}\delta}}{2}\cdot(w_t(x_1,x_2,x_3,x_4) + w_t(y_1,y_2,y_3,y_4))\\
&\quad\leqslant \frac{(1+\epsilon) + (1+\epsilon)^{-1}}{2}\cdot(w_t(x_1,x_2,x_3,x_4) + w_t(y_1,y_2,y_3,y_4))\\
&\quad\leqslant e^{\epsilon^2/2}\,(w_t(x_1,x_2,x_3,x_4) + w_t(y_1,y_2,y_3,y_4))\\
&\quad\leqslant e^{\epsilon^2/2}\,(w_t(x_1,x_2,x_3,x_4) + w_t(y_1,y_2,y_3,y_4)) + \epsilon,
\end{aligned}$$

where the second inequality follows from $(j_1, j_2, j_3, j_4) < (i_1, i_2, i_3, i_4) \implies w_t(x_1, x_2, x_3, x_4) \leqslant w_t(y_1, y_2, y_3, y_4)$ and the rearrangement inequality, and the thir inequality follows from $\frac{1}{3}\delta \leqslant \frac{1}{2} < 1$.

Subcase 2.2: $d_t(y_1, y_2, y_3, y_4) < \delta$.

In this case, $d_{t+1}(y_1, y_2, y_3, y_4) = 0$, and $d_{t+1}(x_1, x_2, x_3, x_4) \leqslant d_t(x_1, x_2, x_3, x_4) + \delta \leqslant d_t(y_1, y_2, y_3, y_4) + \delta < 2\delta$. Then

$$\begin{aligned}
&w_{t+1}(x_1, x_2, x_3, x_4) + w_{t+1}(y_1, y_2, y_3, y_4) \\
&\quad\leqslant (1+\epsilon)^{\frac{2}{3}\cdot 2\delta} + (1+\epsilon)^0 \\
&\quad\leqslant 2 + \epsilon \\
&\quad\leqslant (w_t(x_1, x_2, x_3, x_4) + w_t(y_1, y_2, y_3, y_4)) + \epsilon \\
&\quad\leqslant e^{\epsilon^2/2}\,(w_t(x_1, x_2, x_3, x_4) + w_t(y_1, y_2, y_3, y_4)) + \epsilon.
\end{aligned}$$

◁

Now, using Claim 14 and summing over all bijected tuples $(i_1, i_2, i_3, i_4) \leftrightarrow (j_1, j_2, j_3, j_4)$, we get

$$\psi(t+1) \leqslant e^{\epsilon^2/2} \cdot \psi(t) + \epsilon \cdot \frac{\binom{K}{4}}{2}.$$

Solving the reccurence, we get

$$\begin{aligned}
\psi(t+1) &\leqslant e^{(t+1)\epsilon^2/2} \cdot \binom{K}{4} + \frac{e^{(t+1)\epsilon^2/2} - 1}{e^{\epsilon^2/2} - 1} \cdot \epsilon \cdot \frac{\binom{K}{4}}{2} \\
&\leqslant e^{(t+1)\epsilon^2/2} \cdot \binom{K}{4} \cdot \left(1 + \frac{1}{2(e^{\epsilon^2/2} - 1)}\right) \\
&\leqslant e^{(t+1)\epsilon^2/2} \cdot \binom{K}{4} \cdot \left(1 + \frac{1}{\epsilon}\right).
\end{aligned}$$

Finally, we get our bound on $\frac{1}{2}\mathsf{posc}_\tau(1) + \mathsf{posc}_\tau(2) + \mathsf{posc}_\tau(3) + \mathsf{posc}_\tau(4)$. Let $x_1 = c^{\mathsf{q}}_\tau(1)$, $x_2 = c^{\mathsf{q}}_\tau(2)$, $x_3 = c^{\mathsf{q}}_\tau(3)$, and $x_4 = c^{\mathsf{q}}_\tau(4)$. We have that

$$\begin{aligned}
w_\tau(x_1, x_2, x_3, x_4) = (1+\epsilon)^{\frac{1}{3}\cdot d_\tau(x_1, x_2, x_3, x_4)} &< \psi(\tau) \\
&\leqslant e^{\tau\epsilon^2/2} \cdot \binom{K}{4} \cdot \left(1 + \frac{1}{\epsilon}\right) \\
&\leqslant e^{\tau\epsilon^2/2} \cdot K^4 \cdot \left(1 + \frac{1}{\epsilon}\right),
\end{aligned}$$

which gives

$$\begin{aligned}
\frac{3\tau}{2} - \mathsf{pos}^{\mathsf{q}}_\tau(x_1, x_2, x_3, x_4) &\leqslant d_\tau(x_1, x_2, x_3, x_4) \\
&< \frac{3}{\ln(1+\epsilon)} \cdot \left(\tau\epsilon^2/2 + 4\ln K + \ln\left(1 + \frac{1}{\epsilon}\right)\right) \\
&\leqslant \frac{6}{\epsilon} \cdot \left(n\epsilon^2/2 + 4\ln K + \frac{1}{\epsilon}\right) \\
&\leqslant 3\epsilon n + \frac{6}{\epsilon} \cdot \left(4\ln K + \frac{1}{\epsilon}\right) \\
&\leqslant 5\epsilon n,
\end{aligned}$$

where we use that $\tau \leqslant n$, $\frac{\epsilon}{2} \leqslant \ln(1+\epsilon)$, and $\ln\left(1+\frac{1}{\epsilon}\right) \leqslant 1/\epsilon$, and where the last inequality follows from $n = \frac{24k}{\epsilon^3}$ so that

$$\frac{6}{\epsilon} \cdot \left(4\ln K + \frac{1}{\epsilon}\right) \leqslant \frac{6}{\epsilon}\left(4k + \frac{1}{\epsilon}\right) = \epsilon^2 n + \frac{\epsilon}{4} n < 2\epsilon n.$$

This completes the proof of Lemma 12. ◀

4.3 The $(2, \frac{3}{7} - \epsilon)$-List Decodability of SW

We now return to our main goal of this section, which is to show that SW is 2-list decodable up to radius $\frac{3}{7}$. We restate the theorem below for convenience.

▶ **Theorem 9.** *In the* SW *coin game, no matter which sets Eve chooses each round, it holds that*

$$\mathsf{posc}_n(3) > \left(\frac{3}{7} - 2\epsilon\right) \cdot n.$$

That is, the SW *coin game with* $n = \frac{24k}{\epsilon^3}$ *gives a feedback code that is* $\left(2, \frac{2}{7} - 2\epsilon\right)$*-list-decodable.*

Proof. Consider the first round $n - f$ at which $\mathsf{posc}_{n-f}(4) > \left(\frac{3}{7} - 2\epsilon\right) n$. There must be such a round because $\frac{1}{2}\mathsf{posc}_n(1) + \mathsf{posc}_n(2) + \mathsf{posc}_n(3) + \mathsf{posc}_n(4) \geqslant \frac{3}{2}n - 5\epsilon n$ by Lemma 12, so $\mathsf{posc}_n(4) \geqslant \frac{2}{7} \cdot \left(\frac{3}{2}n - 7\epsilon n\right) = \left(\frac{3}{7} - 2\epsilon\right) n$. After round $n - f$, there only remains 3 relevant coins upon the board. Our goal is to show that another falls off.

Suppose that in round $n - f$ that the first coin is at position $\left(\frac{3}{7} - 2\epsilon\right) n - p$. From Lemma 11, we know that

$$\begin{aligned}
\mathsf{posc}_{n-f}(2) + \mathsf{posc}_{n-f}(3) &\geqslant \mathsf{posc}_{n-f}(1) + \mathsf{posc}_{n-f}(4) - 1 \\
&= \frac{6}{7}n - p - 4\epsilon n - 1 \\
&\geqslant \frac{6}{7}n - p - 5\epsilon n. \qquad (4)
\end{aligned}$$

We also know from Lemma 12 that

$$\begin{aligned}
\mathsf{posc}_{n-f}(2) + \mathsf{posc}_{n-f}(3) &\geqslant \frac{3}{2}(n-f) - 5\epsilon n - \frac{1}{2}\mathsf{posc}_{n-f}(1) - \mathsf{posc}_{n-f}(4) \\
&\geqslant \frac{3}{2}(n-f) - 5\epsilon n - \frac{1}{2}\left(\left(\frac{3}{7} - 2\epsilon\right)n - p\right) - \left(\frac{3}{7} - 2\epsilon\right)n \\
&= \frac{6}{7}n - \frac{3}{2}f + \frac{1}{2}p - 2\epsilon n. \qquad (5)
\end{aligned}$$

Adding $\frac{1}{3}$ of Equation 4 and $\frac{2}{3}$ of Equation 5, we get

$$\mathsf{posc}_{n-f}(2) + \mathsf{posc}_{n-f}(3) \geqslant \frac{6}{7}n - f - 3\epsilon n. \qquad (6)$$

Now, suppose that the third coin does not reach position $\left(\frac{3}{7} - 3\epsilon\right) n$ by the end of the game. Then, at least one of the second and third coins must move in each of the last f rounds, so that $\mathsf{posc}_t(2) + \mathsf{posc}_t(3)$ increases by at least 1 each round. This means that at round n,

$$\begin{aligned}
\mathsf{posc}_n(2) + \mathsf{posc}_n(3) &\geqslant \mathsf{posc}_{n-f}(2) + \mathsf{posc}_{n-f}(3) + f \\
&\geqslant \left(\frac{6}{7} - 3\epsilon\right) n,
\end{aligned}$$

which gives that $\mathsf{posc}_n(3) \geqslant \frac{1}{2}\left(\frac{6}{7} - 3\epsilon\right) n > \left(\frac{3}{7} - 2\epsilon\right) n$, a contradiction on our assumption that the third coin never falls off the board.

Therefore, no matter which sets the adversary chooses, there will always be at most two coins left on the board at the end of the game. ◀

5 Spencer-Winkler Does Not Give $\left(3, \frac{7}{15}\right)$-List Feedback Codes

In Section 4, we showed that the Spencer-Winkler strategy for Alice and Bob achieves a $(2, \frac{3}{7})$-list feedback code. In this strategy, Bob always splits the sets into the even-numbered coins and the odd-numbered coins.

A reasonable conjecture might be that this strategy is always optimal, and, in fact, we do not know of any proof that it is not. However, in this section we will show that for $\ell = 3$, it does not match our upper bound of $\frac{7}{15}$ from Section 3. Whether this is because the protocol is suboptimal or because the upper bound is not tight remains open for further investigation.

▶ **Theorem 3.** *For any $\varepsilon > 0$, the feedback code* SW *(defined in Section 4) is not $\left(3, \frac{31}{67} + \varepsilon\right)$-list decodable for any choice of encoding length $n = n(k)$.*

As in the previous sections, we discuss the counterexample in the coin game formulation. We will show an attack for all large enough k. In this setting, Bob has $K = 2^k$ coins for some $k > 1000\varepsilon^{-1}$, and in each of n rounds (Alice and Bob choose n), he partitions the coins into 2 sets. Then, Eve chooses one of the two sets in which to move all the coins up by 1. Eve wins if, at the end of the protocol, at least 4 coins remain $\lesssim \frac{31}{67}n$.

5.1 Eve's Strategy

We begin by stating Eve's procedure for determining which set of coins to select at each step. For simplicity, we'll assume that n is a multiple of 67 and write $n = 67q$. We'll further assume q is even and εq is an integer.

Protocol 2 : Eve's Procedure

Alice has a coin $x \in \{0,1\}^k$. She and Bob simulate the coin game with $K = 2^k$ coins. For the n rounds, Eve does the following:

1. For the first $32q$ rounds, Eve selects (moves up by 1) the even coins.
2. For the next $16q$ rounds, Eve selects the odd coins.
3. For the next $8q$ rounds, Eve selects the even coins.
4. For the next $4q$ rounds, Eve selects the odd coins.
5. For the next $4q$ rounds, Eve selects the even coins.
6. For the final $3q$ rounds, Eve selects the odd coins.

5.2 Analysis of Eve's Strategy

We will establish our main sequence of claims giving an upper bound on the positions of the first few coins after each of these steps (and a step in between as well). The claims are written out below and summarized by the following table. For each upper bound, we additionally add $+\varepsilon q$, although we do not write it into the table for brevity.

Establishing the bounds in the table would show that $\mathsf{posc}_m(4) \leqslant 31q + \varepsilon q < \left(\frac{31}{67} + \varepsilon\right) \cdot n$ at the end of the n rounds, which establishes Theorem 3.

Now, we will prove the bounds listed in the table. Each argument below will establish the bounds indicated by the corresponding symbol. Each argument requires only that the bounds in the previous line of the table hold, so we can prove the bounds in a top-down order.

	$\mathsf{posc}_m(1)$	$\mathsf{posc}_m(2)$	$\mathsf{posc}_m(3)$	$\mathsf{posc}_m(4)$	$\mathsf{posc}_m(5)$
(even) $m = 32q$	0 (▲)	16q (●)	16q (●)	16q (●)	16q (●)
(odd) $m = 48q$	16q (■)	16q (♦)	24q (●)	24q (●)	24q (●)
(even) $m = 56q$	16q (▲)	24q (■)	24q (♦)	28q (●)	28q (●)
(odd) $m = 60q$	20q (■)	24q (♦)	28q (■)	28q (♦)	30q (●)
(even) $m = 62q$	20q (▲)	26q (■)	28q (♦)	30q (■)	30q (♦)
(even) $m = 64q$	20q (▲)	28q (■)	28q (♦)	31q (♣)	31q (♣)
(odd) $m = 67q$	23q (■)	28q (♦)	31q (■)	31q (♦)	33.5q (●)

Figure 3 An upper bound on the positions of the first 5 coins after m rounds. To each listed position bound, additionally add $+\varepsilon q$. Eve's selection of odd or even is also noted. Symbols in the cells are referenced in the proofs below.

To show the (▲) bounds, note that selecting the even coins never moves the coin in position 1.

Next, we establish the (●) bounds with the following claim. It shows us that for all times m, for $i \leqslant 6$ we have that $\mathsf{posc}_m(i) < \frac{1}{2} \cdot m + \varepsilon q$

▷ Claim 15. For any round m, as long as k is sufficiently large, it holds that $\mathsf{posc}_m(6) < \frac{1}{2} \cdot m + \varepsilon q$.

Proof. Since at all steps, only the odd coins or even coins move (and there are an even number of coins total), it holds that

$$\mathsf{posc}_m(1) + \ldots + \mathsf{posc}_m(K) \leqslant \frac{1}{2} \cdot Km$$

Then,

$$\frac{1}{2} \cdot Km > \mathsf{posc}_m(6) + \ldots + \mathsf{posc}_m(K) \geqslant (K-6) \cdot \mathsf{posc}_m(6)$$

which implies that

$$\mathsf{posc}_m(6) < \frac{1}{2} \cdot \frac{Km}{K-6} < \frac{1}{2} \cdot m + \varepsilon q$$

where the last line holds if $K = 2^k$ is sufficiently large, for example larger than $1000\varepsilon^{-1}$. ◁

To show the (■) bounds, note the following claim.

▷ Claim 16. For all $i \in [K]$, between rounds t and t', it holds that $\mathsf{posc}_{t'}(i) - \mathsf{posc}_t(i) \leqslant t' - t$.

Proof. All of coins $c_t(1), \ldots, c_t(i)$ must have have position at most $\mathsf{posc}_t(i)$ after round t. They can all move at most $t' - t$ by round t', so there are at least t' coins at position $\mathsf{posc}_t(i) + (t' - t)$ or earlier by the end of round t', which is what we desired. ◁

Next, we establish the (♦) bounds with the following claim.

▷ Claim 17. For all coins $i \in [K-1]$ and times t and t', if the set corresponding to parity $i+1$ is not chosen in that interval, then $\mathsf{posc}_{t'}(i+1) \leqslant \max\{\mathsf{posc}_t(i+1), \mathsf{posc}_t(i) + t' - t\}$.

Proof. Let $d := \mathsf{posc}_t(i+1) - \mathsf{posc}_t(i)$. For the first d steps then, the coins $c_t(1), \ldots, c_t(i)$ are all necessarily before the coin $c_t(i+1)$. Then, $\mathsf{posc}_t(i+1)$ does not increase for those d rounds (if $t'-t < d$, then simply stop at round t'. Then, by Claim 16, in the remaining $\max\{t'-t-d, 0\}$ rounds, $\mathsf{posc}_t(i+1)$ increases by at most 1. Therefore, in total, the final position of $c_t(i+1)$ after round t' is $\max\{t'-t-d+\mathsf{posc}_t(i+1), \mathsf{posc}_t(i+1)\}$. is This gives the desired bound because $t'-t-d+\mathsf{posc}_t(i+1) \leqslant t'-t+\mathsf{posc}_t(i)-\mathsf{posc}_t(i+1)+\mathsf{posc}_t(i+1) = t'-t+\mathsf{posc}_t(i)$ as desired. ◁

Finally, we establish the (♣) bounds specifically for $\mathsf{posc}_m(4)$ and $\mathsf{posc}_m(5)$ between rounds $62q$ and $64q$.

▷ Claim 18. After round $m+d$ where $m = 62q$ and d is even and $d \leqslant 2q$, it holds that $\mathsf{posc}_{m+d}(5) \leqslant 30q + d/2 + \varepsilon q$.

Proof. We will show this statement inductively. This holds when $d = 0$ as the base case.

Assume this is true for d, and we'll prove it for $d+2$. By Claim 16, $\mathsf{posc}(5)$ can increase by at most 2 in these 2 steps, so we only have to address the case where $\mathsf{posc}_{m+d}(5) = 30q+d/2+\varepsilon q$ exactly. Moreover, we must have $\mathsf{posc}_{m+d}(4) = 30q+d/2+\varepsilon q$ or $\mathsf{posc}_{m+d}(4) = 30q+d/2+\varepsilon q$ exactly, otherwise we would be done by Claim 17.

All the coins $c_m(1), c_m(2), c_m(3)$ are in position at most $28q+d+\varepsilon q < 30q+d/2+\varepsilon q-10$ at the end of d steps, so they cannot surpass $c_{m+d}(4)$ during these next 2 steps.

In both the cases where $\mathsf{posc}_{m+d}(4) = 30q + d/2 + \varepsilon q$ and $\mathsf{posc}_{m+d}(4) = 30q + d/2 + \varepsilon q$, both the 4th and 5th positions move up by exactly 1 if the first three coins are guaranteed to be out of the picture. ◁

References

1 Rudolf Ahlswede, Ferdinando Cicalese, Christian Deppe, and Ugo Vaccaro. Two batch search with lie cost. *IEEE transactions on information theory*, 55(4):1433–1439, 2009. doi:10.1109/TIT.2009.2013014.

2 Omar Alrabiah, Venkatesan Guruswami, and Ray Li. Randomly punctured reed–solomon codes achieve list-decoding capacity over linear-sized fields. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1458–1469, 2024. doi:10.1145/3618260.3649634.

3 Elwyn R. Berlekamp. *Block coding with noiseless feedback*. PhD thesis, Massachusetts Institute of Technology, 1964.

4 Elwyn R. Berlekamp. Block coding for the binary symmetric channel with noiseless, delayless feedback. *Error-correcting Codes*, pages 61–88, 1968.

5 Volodia M Blinovskiı. Bounds for codes in decoding by a list of finite length. *Problemy Peredachi Informatsii*, 22(1):11–25, 1986.

6 Joshua Brakensiek, Sivakanth Gopi, and Visu Makam. Generic reed-solomon codes achieve list-decoding capacity. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1488–1501, 2023. doi:10.1145/3564246.3585128.

7 Christian Deppe. Solution of ulam's searching game with three lies or an optimal adaptive strategy for binary three-error-correcting codes. *Discrete Mathematics*, 224(1-3):79–98, 2000. doi:10.1016/S0012-365X(00)00109-6.

8 Christian Deppe. Coding with feedback and searching with lies. In *Entropy, Search, Complexity*, pages 27–70. Springer, 2007.

9 Peter Elias. *List decoding for noisy channels*. Massachusetts Institute of Technology, 1957.

10 Ran Gelles. Coding for Interactive Communication: A Survey. *Foundations and Trends® in Theoretical Computer Science*, 13:1–161, January 2017. doi:10.1561/0400000079.

11 Zeyu Guo and Zihan Zhang. Randomly punctured reed-solomon codes achieve the list decoding capacity over polynomial-size alphabets. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 164–176. IEEE, 2023. doi:10.1109/FOCS57990.2023.00019.

12 Meghal Gupta, Venkatesan Guruswami, and Rachel Yun Zhang. Binary error-correcting codes with minimal noiseless feedback. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pages 1475–1487, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3564246.3585126.

13 Venkatesan Guruswami. *List decoding of error correcting codes*. PhD thesis, Massachusetts Institute of Technology, 2001. URL: https://api.semanticscholar.org/CorpusID:9164141.

14 Venkatesan Guruswami. List decoding of binary codes–a brief survey of some recent results. In *Coding and Cryptology: Second International Workshop, IWCC 2009, Zhangjiajie, China, June 1-5, 2009. Proceedings 2*, pages 97–106. Springer, 2009. doi:10.1007/978-3-642-01877-0_10.

15 Venkatesan Guruswami. Linear-algebraic list decoding of folded reed-solomon codes. In *2011 IEEE 26th Annual Conference on Computational Complexity*, pages 77–85. IEEE, 2011. doi:10.1109/CCC.2011.22.

16 Venkatesan Guruswami et al. Algorithmic results in list decoding. *Foundations and Trends® in Theoretical Computer Science*, 2(2):107–195, 2007.

17 Venkatesan Guruswami, Ray Li, Jonathan Mosheiff, Nicolas Resch, Shashwat Silas, and Mary Wootters. Bounds for list-decoding and list-recovery of random linear codes. *arXiv preprint arXiv:2004.13247*, 2020. arXiv:2004.13247.

18 Venkatesan Guruswami and Srivatsan Narayanan. Combinatorial limitations of average-radius list-decoding. *IEEE Transactions on Information Theory*, 60(10):5827–5842, 2014. doi:10.1109/TIT.2014.2343224.

19 Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Transactions on information theory*, 54(1):135–150, 2008. doi:10.1109/TIT.2007.911222.

20 Wojciech Guzicki. Ulam's searching game with two lies. *Journal of Combinatorial Theory, Series A*, 54(1):1–19, 1990. doi:10.1016/0097-3165(90)90002-E.

21 Bernhard Haeupler, Pritish Kamath, and Ameya Velingker. Communication with Partial Noiseless Feedback. In *APPROX-RANDOM*, 2015.

22 S Muthukrishnan. On optimal strategies for searching in presence of errors. In *Proceedings of the fifth annual ACM-SIAM symposium on discrete algorithms*, pages 680–689, 1994. URL: http://dl.acm.org/citation.cfm?id=314464.314672.

23 Alberto Negro, Giuseppe Parlati, and P Ritrovato. Optimal adaptive search: reliable and unreliable models. In *Proc. 5th Italian Conf. on Theoretical Computer Science*, pages 211–231. World Scientific, 1995.

24 Andrzej Pelc. Solution of ulam's problem on searching with a lie. *Journal of Combinatorial Theory, Series A*, 44(1):129–140, 1987. doi:10.1016/0097-3165(87)90065-3.

25 Andrzej Pelc. Searching games with errors—fifty years of coping with liars. *Theoretical Computer Science*, 270(1-2):71–109, 2002. doi:10.1016/S0304-3975(01)00303-6.

26 Morris Plotkin. Binary codes with specified minimum distance. *IRE Transactions on Information Theory*, 6(4):445–450, 1960. doi:10.1109/TIT.1960.1057584.

27 A. Renyi. On a problem of information theory. In *MTA Mat. Kut. Int. Kozl.*, volume 6B, pages 505–516, 1961.

28 Nicolas Resch, Chen Yuan, and Yihan Zhang. Zero-rate thresholds and new capacity bounds for list-decoding and list-recovery. *IEEE Transactions on Information Theory*, 2024.

29 Eric Ruzomberka, Yongkyu Jang, David J. Love, and H. Vincent Poor. The capacity of channels with o(1)-bit feedback. In *2023 IEEE International Symposium on Information Theory (ISIT)*, pages 2768–2773, 2023. doi:10.1109/ISIT54713.2023.10206714.

30 Ofer Shayevitz. On error correction with feedback under list decoding. In *IEEE International Symposium on Information Theory, ISIT 2009, June 28 - July 3, 2009, Seoul, Korea, Proceedings*, pages 1253–1257, August 2009. doi:10.1109/ISIT.2009.5205965.

31 Joel Spencer and Peter Winkler. Three Thresholds for a Liar. *Combinatorics, Probability and Computing*, 1(1):81–93, 1992. doi:10.1017/S0963548300000080.

32 Madhu Sudan. List decoding: Algorithms and applications. *ACM SIGACT News*, 31(1):16–27, 2000. doi:10.1145/346048.346049.

33 Stanislaw M Ulam. *Adventures of a Mathematician*. Univ of California Press, 1991.

34 John M. Wozencraft. *List decoding*, volume 48. Massachusetts Institute of Technology, 1958.

35 K.Sh. Zigangirov. Number of correctable errors for transmission over a binary symmetrical channel with feedback. *Problems Inform. Transmission*, 12:85–97, 1976.

Hardness of Sampling for the Anti-Ferromagnetic Ising Model on Random Graphs

Neng Huang
University of Michigan, Ann Arbor, MI, USA

Will Perkins
Georgia Institute of Technology, Atlanta, GA, USA

Aaron Potechin
University of Chicago, IL, USA

Abstract

We prove a hardness of sampling result for the anti-ferromagnetic Ising model on random graphs of average degree d for large constant d, proving that when the normalized inverse temperature satisfies $\beta > 1$ (asymptotically corresponding to the condensation threshold), then w.h.p. over the random graph there is no stable sampling algorithm that can output a sample close in W_2 distance to the Gibbs measure. The results also apply to a fixed-magnetization version of the model, showing that there are no stable sampling algorithms for low but positive temperature max and min bisection distributions. These results show a gap in the tractability of search and sampling problems: while there are efficient algorithms to find near optimizers, stable sampling algorithms cannot access the Gibbs distribution concentrated on such solutions.

Our techniques involve extensions of the interpolation technique relating behavior of the mean field Sherrington-Kirkpatrick model to behavior of Ising models on random graphs of average degree d for large d. While previous interpolation arguments compared the free energies of the two models, our argument compares the average energies and average overlaps in the two models.

2012 ACM Subject Classification Mathematics of computing → Probability and statistics; Theory of computation → Randomness, geometry and discrete structures

Keywords and phrases Random graph, spin glass, sampling algorithm

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.61

Related Version *Full Version*: `https://arxiv.org/abs/2409.03974`

Funding *Neng Huang*: Work was primarily done when NH was affiliated with the University of Chicago, supported partly by the Institute for Data, Econometrics, Algorithms, and Learning (IDEAL) with NSF grant ECCS-2216912.
Will Perkins: Supported in part by NSF grant CCF-2309708.

1 Introduction

The study of disordered systems is at the intersection of statistical physics, probability theory, and computer science. Models of disordered systems, such as the Edwards–Anderson model, Sherrington–Kirkpatrick model, random constraint satisfaction problems, and combinatorial optimization problems on random networks, have been studied from many different perspectives, including as toy models of complex physical systems and as a source of random computational problems on which to test algorithms and understand the sources of algorithmic intractability.

In the algorithmic context, disordered systems present at least two natural algorithmic challenges: the first is the *search problem*, that of finding a high quality solution to an optimization problem or finding a near ground state in the language of statistical physics; the second is the *sampling problem*, that of (approximately) sampling from a Gibbs measure that weights solutions exponentially by their objective value.

© Neng Huang, Will Perkins, and Aaron Potechin;
licensed under Creative Commons License CC-BY 4.0

One striking phenomenon that has been discovered is a gap between the tractability of search and sampling. That is, for certain choices of parameters in some models of disordered systems, the search problem is tractable (there exist polynomial-time or even near-linear time algorithms to find high quality solutions), while no efficient sampling algorithms are known, and in fact broad classes of sampling algorithms can be proved to be ineffective. So while high quality solutions can be found efficiently, these solutions are not typical, in the sense that the distribution of solutions found by a given search algorithm is far from the equilibrium Gibbs measure.

A prime example of this phenomenon occurs in the Sherrington–Kirkpatrick (SK) model [36], an Ising model on the complete graph on n vertices in which the $\binom{n}{2}$ coupling constants are independent Gaussian random variables (see e.g [32] for a mathematical survey of the model). Here, after appropriate normalization, the key parameter is β, the inverse temperature of the model. Parisi famously predicted a formula for the limiting free energy of the SK model as a function of β [33, 34]; this formula is non-analytic at $\beta = 1$, indicating the existence of the phase transition at this point. The Parisi formula was proved rigorously by Guerra [23] and Talagrand [39], with methods that have proved very influential in mathematics, physics, and computer science in the past two decades. Along with the Parisi formula comes information about the SK Gibbs measure itself; in particular, Parisi identified the "average overlap" of two independent samples from the Gibbs measure as an order parameter for the phase transition [35] and developed the theory of a hierarchy of "replica symmetry breaking". Thus the equilibrium picture of the SK model is fairly well understood, with the phase transition identified at $\beta = 1$.

Turning to the algorithmic search problem, the task is to find a solution $\sigma \in \{\pm 1\}^n$ that (approximately) maximizes the probability mass function of the SK model; that is, the task of finding a ground state or near ground state of the model. Montanari [30], partly inspired by the work of Subag on spherical spin glasses [38], proposed an algorithm for this task based on approximate message passing, which finds an approximate optimizer under the widely believed assumption that the SK model has "no overlap gap" (sometimes also referred to as "full replica symmetry breaking") for large enough β. This approach was later extended to mixed p-spin models [3], which can be thought of as the generalization of the Ising model to hypergraphs.

For the sampling problem, however, the picture is less optimistic. It was predicted by physicists that the simple Glauber dynamics should converge fast as long as $\beta < 1$ [37], but progress on a rigorous proof of this statement has only come recently. A series of recent works [7, 22, 18, 5] showed that Glauber dynamics does indeed converge quickly if $\beta < 1/4$ (more specifically, in $O(n \log n)$ steps). A more recent work [6] has further extended the range of rapid convergence to values beyond $\beta = 1/4$. Using a different approach based on *stochastic localization*, El Alaoui, Montanari, and Sellke [4] gave a different sampling algorithm that approximately samples from the SK Gibbs measure for $\beta < 1/2$, with the approximation in terms of the Wasserstein distance instead of the total variation distance guarantees given by previous works. This result was later improved to the entire replica-symmetric phase $\beta < 1$ by Celentano [11]. On the other hand, obstacles seem to arise after the phase transition at $\beta = 1$. El Alaoui, Montanari, and Sellke showed that when $\beta > 1$, the onset of *disorder chaos* naturally obstructs *stable* sampling algorithms [4]. Similar ideas were used recently by El Alaoui and Gamarnik to rule out stable sampling algorithms for the symmetric binary perceptron problem [1].

One can ask whether this gap in tractability for the search and sampling problems is universal for certain classes of disordered systems. The SK model is an example of a *mean-field model*: all spins (variables) interact with each other. Mean-field models have the

advantage of being mathematically tractable to some degree, but the disadvantage of being unrealistic from the physics perspective as well as the perspective of large real-world networks. Seeking a trade-off between these two aspects, physicists have studied *diluted mean-field models*: that is, statistical physics models on random graphs or hypergraphs of constant average degree [29]. These models inherit some of the symmetries of mean-field models while also having some non-trivial local geometry. The study of diluted mean field models in physics led to rich and surprising predictions for the behavior of optimization problems on random graphs and random constraint satisfaction problems [27, 25, 20]. Rigorously proving these predictions is a major task in mathematics and theoretical computer science.

The specific diluted mean-field model we address here is the anti-ferromagnetic Ising model on sparse random graphs. The Hamiltonian of this model is the number of edges in the random graph cut by the given configuration, and the ground states correspond to the maximum cuts in the graph. Finding a maximum cut in a graph is one of the best-known constraint satisfaction problems, and due to this connection this model has been studied extensively in computer science as well as statistical physics. It is known that as the average degree of the random graph increases, many aspects of this model can be understood via the SK model. Dembo, Montanari, and Sen showed that the typical size of a maximum cut in both Erdős-Rényi random graphs and random regular graphs can be related to the ground state energy of SK model [21]. They proved this by interpolating the free energy between the two models (described in more detail below). El Alaoui, Montanari, and Sellke then gave an algorithm for finding a near maximum cut in a random regular graph by adapting Montanari's message-passing algorithm for the SK model [2]. Similar results have also been shown for the more general mixed p-spin models and their diluted counterparts [24, 13].

It is natural to ask if this connection between the anti-ferromagnetic Ising model and the SK model can be extended further. In particular, does the search vs. sampling phenomenon also arise in the anti-ferromagnetic Ising model on random graphs? In this paper, we give an affirmative answer to this question.

1.1 Main Results

Before stating our main results we introduce some necessary definitions and notation. We refer to an element in $\{-1,1\}^n$ as a *configuration*. Given a graph G and its adjacency matrix A_G, we define the *Ising Hamiltonian* $H_G(\sigma) = \sum_{1\le i\le j\le n}(A_G)_{ij}\sigma(i)\sigma(j)$. Note that $\frac{|E(G)|+H_G(\sigma)}{2}$ is exactly the number of edges in the graph whose endpoints are assigned the same spin by σ. In particular, a ground state (i.e., a configuration that minimizes the Hamiltonian) of H_G corresponds to a maximum cut in the graph. For any *inverse temperature parameter* $\beta \in \mathbb{R}$, H_G induces the *Gibbs distribution*

$$\mu_{\beta,G}(\sigma) = \frac{\exp(-\beta H(\sigma))}{Z_G(\beta)}.$$

where

$$Z_G(\beta) = \sum_{\sigma'\in\{-1,1\}^n} \exp(-\beta H(\sigma'))$$

is the *partition function*, the normalizing constant of the Gibbs distribution.

In this paper, as is common in prior works, the random graph model we consider will be the Poissonized random multigraph model $\mathbb{G}^{\mathrm{Po}}_{n,d}$, where we first sample the number of edges from a Poisson distribution with mean $dn/2$ (so that the expected average degree is d), and then for each edge sample both of its endpoints independently from $[n] = \{1, 2, \dots, n\}$.

Given any two measures μ_1, μ_2 over $\{-1,1\}^n$, we define their *normalized Wasserstein distance* to be

$$W_{2,n}(\mu_1,\mu_2) = \inf_{\pi\in\Gamma(\mu_1,\mu_2)} \left(\mathop{\mathbf{E}}_{(\sigma_1,\sigma_2)\sim\pi} \left[\frac{1}{n}\sum_{i=1}^{n} (\sigma_1(i)-\sigma_2(i))^2 \right] \right)^{1/2},$$

where $\Gamma(\mu_1,\mu_2)$ is the set of all couplings of μ_1 and μ_2; that is, measures over $\{-1,1\}^n \times \{-1,1\}^n$ whose marginal on the first argument is μ_1 and the second μ_2.

Our main result is a hardness of sampling theorem in terms of the $W_{2,n}$ distance for the measure $\mu_{\beta,G}$ against *stable* sampling algorithms where $G \sim \mathbb{G}^{\mathrm{Po}}_{n,d}$. Informally, a sampling algorithm is stable if its output is insensitive to small perturbations of the input. See Definition 12 for a formal definition.

▶ **Theorem 1.** *Let $\{\mathsf{ALG}_n\}_{n\geq 1}$ be a family of randomized sampling algorithms where ALG_n takes as input an n-vertex (multi-)graph G and an inverse temperature parameter β and produces an output law $\mu^{\mathsf{ALG}}_{\beta,G}$ over $\{-1,1\}^n$. For every $\beta > 1$ and d sufficiently large, if $\{\mathsf{ALG}_n\}_{n\geq 1}$ is* stable *at inverse temperature parameter $\beta/\sqrt{d}$, then*

$$\liminf_{n\to\infty} \mathop{\mathbf{E}}_{G\sim\mathbb{G}^{\mathrm{Po}}_{n,d}} \left[W_{2,n}\left(\mu^{\mathsf{ALG}}_{\beta/\sqrt{d},G}, \mu_{\beta/\sqrt{d},G}\right)\right] > 0.$$

That is, stable algorithms cannot sample from the model with vanishing error in Wasserstein distance.

It is known that the replica symmetry breaking threshold for $\mathbb{G}^{\mathrm{Po}}_{n,d}$ occurs at the Kesten-Stigum bound $\beta^\dagger(d) = \frac{1}{2}\log\frac{\sqrt{d}+1}{\sqrt{d}-1}$[1] [31, 19], so the threshold $\beta = 1$ corresponds exactly to the normalized limit $\lim_{d\to\infty}\sqrt{d}\cdot\beta^\dagger(d) = 1$. Note that $\beta = 1$ is also the replica symmetry breaking threshold of the Sherrington-Kirkpatrick model.

As it turns out, the same proof for Theorem 1 yields sampling hardness for near-maximum and near-minimum bisections as well. Let $A_n = \{\sigma\in\{-1,1\}^n : |\sum_{i=1}^n \sigma(i)| \leq 1\}$ be the set of configurations in which the numbers of $+1$s and -1s differ by at most one, which we refer to as *bisections*. The Gibbs distribution $\mu_{\beta,G}$ when restricted to A_n gives

$$\mu^{\mathsf{bis}}_{\beta,G}(\sigma) = \frac{\exp(-\beta H(\sigma))}{Z^{\mathsf{bis}}_G(\beta)}.$$

where

$$Z^{\mathsf{bis}}_G(\beta) = \sum_{\sigma'\in A_n} \exp(-\beta H(\sigma'))$$

is the bisection partition function. Note that $\mu^{\mathsf{bis}}_{\beta,G}(\sigma)$ prefers bisections that cut more edges when $\beta > 0$, and bisections with fewer edges when $\beta < 0$. This model is also known as the zero-magnetization Ising model and has been studied both in the statistical physics literature on disordered systems [28] and recently in computer science [10, 8, 26].

▶ **Theorem 2.** *Let $\{\mathsf{ALG}_n\}_{n\geq 1}$ be a family of randomized sampling algorithms where ALG_n takes as input an n-vertex (multi-)graph G and an inverse temperature parameter β and produces an output law $\mu^{\mathsf{ALG}}_{\beta,G}$ over $\{-1,1\}^n$. For every $\beta\in\mathbb{R}$ such that $|\beta| > 1$ and d sufficiently large, if $\{\mathsf{ALG}_n\}_{n\geq 1}$ is* stable *at inverse temperature parameter $\beta/\sqrt{d}$, then*

$$\liminf_{n\to\infty} \mathop{\mathbf{E}}_{G\sim\mathbb{G}^{\mathrm{Po}}_{n,d}} \left[W_{2,n}\left(\mu^{\mathsf{ALG}}_{\beta/\sqrt{d},G}, \mu^{\mathsf{bis}}_{\beta/\sqrt{d},G}\right)\right] > 0.$$

[1] Unless otherwise specified, all logarithms in this paper are natural logarithms.

Theorems 1 and 2 exhibit a gap between search and sampling for the maximum cut and max/minimum bisection problem: while we have a search algorithm known under the widely believed "No Overlap Gap" conjecture [2] [2], no stable algorithm can sample from the Gibbs distribution with arbitrary precision in the Wasserstein metric. Put another way, the algorithm of [2] finds solutions of value $(1-\epsilon)\cdot \mathrm{OPT}$ for any fixed $\epsilon > 0$; Theorems 1 and 2 rule out stable algorithms that sample solutions of these values approximately uniformly, since a standard reduction of counting to sampling (e.g [9]) reduces the task of sampling from the Gibbs distribution to sampling uniformly from solutions of a given value.

1.2 Techniques

To show that sampling from the anti-ferromagnetic Ising Gibbs distribution on random graphs is hard for stable sampling algorithms, we use the framework that was used by [4] to show that sampling from the Sherrington-Kirkpatrick Gibbs distribution is hard. The key quantity in this approach is the *overlap* between two configurations σ_1, σ_2, defined by

$$R_{1,2}(\sigma_1,\sigma_2) := \frac{1}{n}\sum_{i=1}^{n}\sigma_1(i)\sigma_2(i)\,.$$

The framework consists of the following two steps:

1. Show that w.h.p. over the disorder, the average of the squared overlap $R_{1,2}(\sigma_1,\sigma_2)^2$ with respect to two independent samples σ_1 and σ_2 from the Gibbs distribution is at least some positive constant independent of n. In particular, this is expected to hold when the Gibbs distribution exhibits replica symmetry breaking, which explains the threshold $\beta = 1$.
2. Show that the Gibbs distribution exhibits *disorder chaos*. That is, if we perturb the input (the random coupling constants in the case of the SK model or the random graph in the case of the Ising model) slightly (see Section 2 for a formal definition of this perturbation) and then take a sample σ_1 from the Gibbs distribution for the original system and a sample σ_2 from the Gibbs distribution for the perturbed system, with high probability $R_{1,2}(\sigma_1,\sigma_2)$ is very close to zero.

Combining these two steps and using a connection between $R_{1,2}$ and $W_{2,n}$ distance (see Lemma 15), we have that for arbitrarily small perturbations, the $W_{2,n}$ distance between the original Gibbs distribution and the perturbed Gibbs distribution is at least some positive constant. However, for stable sampling algorithms, if we make the perturbation sufficiently small then the $W_{2,n}$ distance between the old and new output distributions can also be arbitrarily small. This implies that the output distribution of any stable sampling algorithm will have strictly positive $W_{2,n}$ distance from the target Gibbs distribution.

In order to carry out this framework for sparse random graph models, we prove new *structural properties* of the Gibbs measures $\mu_{\beta/\sqrt{d},G}$ and $\mu^{\mathsf{bis}}_{\beta/\sqrt{d},G}$ and connect them to those of the Gibbs measures $\mu_{\beta,\mathbf{g}}$ and $\mu^{\mathsf{bis}}_{\beta,\mathbf{g}}$ of the SK model (see Section 2 for the formal definition of $\mu_{\beta,\mathbf{g}}$ and $\mu^{\mathsf{bis}}_{\beta,\mathbf{g}}$). This further extends the close connection between the behavior of the SK model and the sparse random graph model. Previously, it was shown by Dembo, Montanari, and Sen [21] that the *free energy* of the sparse model converges to that of the SK model as the average degree d increases to infinity. We first extend this to the average Hamiltonian of

[2] [2] stated the result for random regular graphs, but it can be transferred to other graph models using e.g. the argument in [13].

a sample drawn from the Gibbs distribution, showing that this quantity also converges in the same manner, regardless of the bisection restriction. This is done by using convexity of the free energy and a well-known observation that the average Hamiltonian can be obtained by taking derivative of the free energy (see (3.1) and (3.2)). The main difficulty here is establishing that the limiting free energy is the same regardless of the bisection restriction, the details of which can be found in the full version of this paper.

▶ **Proposition 3.**

(a) *For every $\beta \geq 0$, we have*

$$\lim_{d\to\infty} \limsup_{n\to\infty} \frac{1}{n} \left| \mathop{\mathbf{E}}_{\mathbf{g}} \left[\mathop{\mathbf{E}}_{\sigma\sim\mu_{\beta,\mathbf{g}}} [H_{\mathrm{SK}}(\sigma)] \right] - \frac{1}{\sqrt{d}} \mathop{\mathbf{E}}_{G} \left[\mathop{\mathbf{E}}_{\sigma\sim\mu_{\beta/\sqrt{d},G}} [H_G(\sigma)] \right] \right| = 0.$$

(b) *For every $\beta \in \mathbb{R}$, we have*

$$\lim_{d\to\infty} \limsup_{n\to\infty} \frac{1}{n} \left| \mathop{\mathbf{E}}_{\mathbf{g}} \left[\mathop{\mathbf{E}}_{\sigma\sim\mu^{\mathsf{bis}}_{\beta,\mathbf{g}}} [H_{\mathrm{SK}}(\sigma)] \right] - \frac{1}{\sqrt{d}} \mathop{\mathbf{E}}_{G} \left[\mathop{\mathbf{E}}_{\sigma\sim\mu^{\mathsf{bis}}_{\beta/\sqrt{d},G}} [H_G(\sigma)] \right] \right| = 0.$$

Our main technical contribution is the following proposition which says that as the average degree d goes to infinity, the expected average squared overlap between two independent samples from the Gibbs measure $\mu_{\beta/\sqrt{d},G}$ converges to those sampled from the Gibbs measure $\mu_{\beta,\mathbf{g}}$ of the SK model, again regardless of the bisection restriction. This gives the first ingredient in the sampling hardness framework. For the SK Hamiltonian, there is a known connection between the average Hamiltonian and the average squared overlap, which can be established using the Gaussian integration by parts trick (see e.g. [32] for more details). In this paper, we show that one can use the Stein-Chen identity for the Poisson distribution (see Lemma 24) in place of the Gaussian integration by parts trick to establish a similar connection for the sparse model in the large degree limit.

▶ **Proposition 4.**

(a) *For any $\beta \geq 0$,*

$$\lim_{d\to\infty} \limsup_{n\to\infty} \left| \mathop{\mathbf{E}}_{\mathbf{g}} \left[\mathop{\mathbf{E}}_{\sigma_1,\sigma_2\sim\mu_{\beta,\mathbf{g}}} \left[R_{1,2}(\sigma_1,\sigma_2)^2 \right] \right] - \mathop{\mathbf{E}}_{G} \left[\mathop{\mathbf{E}}_{\sigma_1,\sigma_2\sim\mu_{\beta/\sqrt{d},G}} \left[R_{1,2}(\sigma_1,\sigma_2)^2 \right] \right] \right| = 0.$$

(b) *For any $\beta \in \mathbb{R}$,*

$$\lim_{d\to\infty} \limsup_{n\to\infty} \left| \mathop{\mathbf{E}}_{\mathbf{g}} \left[\mathop{\mathbf{E}}_{\sigma_1,\sigma_2\sim\mu^{\mathsf{bis}}_{\beta,\mathbf{g}}} \left[R_{1,2}(\sigma_1,\sigma_2)^2 \right] \right] - \mathop{\mathbf{E}}_{G} \left[\mathop{\mathbf{E}}_{\sigma_1,\sigma_2\sim\mu^{\mathsf{bis}}_{\beta/\sqrt{d},G}} \left[R_{1,2}(\sigma_1,\sigma_2)^2 \right] \right] \right| = 0.$$

Finally, to obtain disorder chaos for the sparse model which is the second component in the sampling hardness framework, we use the fact that two configurations sampled from the coupled system for the SK model have nontrivial overlap with only exponentially small probability (see Theorem 29). This causes a gap in free energy between the coupled system and uncoupled system, which can then be transferred to the sparse model using the interpolation result by [17, 16]. We can then translate this gap back to a disorder chaos statement.

1.3 Organization

The paper is organized as follows. In Section 2 we set up the notation and give a detailed overview of the proof. In Section 3 and Section 4, we establish the correspondence of average energy and average overlap between the two models. Finally in Section 5, we transfer the disorder chaos property from the SK model to the sparse random graph models.

2 Overview of the proof

Gibbs distributions

For $\sigma \in \{-1,1\}^n, \mathbf{X} \in \mathbb{R}^{n\times n}$, we define the Hamiltonian

$$H(\sigma;\mathbf{X}) := \sum_{i,j=1}^{n} \mathbf{X}_{ij}\sigma(i)\sigma(j).$$

For any $\beta \in \mathbb{R}$, it induces the following Gibbs distribution on $\{-1,+1\}^n$:

$$\mu_{\beta,\mathbf{X}}(\sigma) = \frac{\exp\left(-\beta H(\sigma;\mathbf{X})\right)}{Z(\beta,\mathbf{X})},$$

where $Z(\beta,\mathbf{X}) = \sum_{\sigma\in\{-1,1\}^n} \exp\left(-\beta H(\sigma;\mathbf{X})\right)$ is the normalizing factor commonly referred to as the partition function. β is commonly referred to as the *inverse temperature* parameter in statistical physics. We also consider the restriction of the above distribution to bisections $\sigma \in A_n$, with

$$\mu^{\mathsf{bis}}_{\beta,\mathbf{X}}(\sigma) = \frac{\exp\left(-\beta H(\sigma;\mathbf{X})\right)}{Z^{\mathsf{bis}}(\beta,\mathbf{X})}, \quad Z^{\mathsf{bis}}(\beta,\mathbf{X}) = \sum_{\sigma\in A_n} \exp\left(-\beta H(\sigma;\mathbf{X})\right).$$

We refer to such restricted versions as bisection models. In statistical physics literature, this is sometimes also called the model with *zero magnetization* and the version without the bisection restriction is called the model with *non-fixed magnetization*. Here, the term magnetization refers to the quantity $m(\sigma) = \sum_{i=1}^{n} \sigma(i)/n$. We sometimes also write $\mu(\sigma;\beta,\mathbf{X})$ and $\mu^{\mathsf{bis}}(\sigma;\beta,\mathbf{X})$ instead of $\mu_{\beta,\mathbf{X}}(\sigma)$ and $\mu^{\mathsf{bis}}_{\beta,\mathbf{X}}(\sigma)$ if we wish to stress the dependence on β and $\mathbf{X}$.

$\mathbf{X}$ is a random matrix drawn from some distribution and in this paper we consider two important cases. In the first case, $\mathbf{X} = \mathbf{g}$, where $\mathbf{g}_{ij} \sim N(0,1/2n)$ independently. This is known as the Sherrington-Kirkpatrick model. In the second case we have $\mathbf{X} = \mathbf{A}$, where each entry $\mathbf{A}_{ij}$ is an independent $\mathrm{Po}(d/2n)$ random variable for some parameter $d > 0$. This case corresponds to sparse random (multi-)graphs with average degree d, where the vertex set is $[n] = \{1,2,\ldots,n\}$ and the multiplicity of the edge $\{i,j\}$ is $\mathbf{A}_{ij} + \mathbf{A}_{ji}$ if $i \neq j$ and $\mathbf{A}_{ii}$ if $i = j$ (note that here $\mathbf{A}$ is not the adjacency matrix of the graph and can be asymmetric). Any configuration σ can be thought of as the indicator vector for a cut where the vertices assigned 1 by σ is on one side and those assigned -1 is on the other side, and $H(\sigma;\mathbf{A})$ is equal to the difference between the number of edges crossing the cut and the number of edges not crossing the cut. When $\beta > 0$, the Gibbs measure prefers those configurations that cut more edges, and this corresponds to the *Maximum Cut* problem for the non-fixed magnetization model and the *Maximum Bisection* problem for the zero-magnetization model. This case is known as the anti-ferromagnetic Ising model in the statistical physic literature. When $\beta < 0$, the non-fixed magnetization model corresponds to the *Minimum Cut* problem while the zero-magnetization model corresponds to the *Minimum Bisection* problem. This case is known as the ferromagnetic Ising model in the statistical physic literature.

For notational convenience, we sometimes drop the random matrix and write $H_{\mathrm{SK}}(\sigma) := H(\sigma;\mathbf{g})$ and $H_d(\sigma) := H(\sigma;\mathbf{A})$. We sometimes refer to these two models as *dense* or *sparse* models respectively.

Gibbs average

For any function $f : \{-1,1\}^n \to \mathbb{R}$, we can define its average with respect to μ or μ^{bis}:

$$\langle f(\sigma)\rangle_{\beta,\mathbf{X}} = \sum_{\sigma\in\{-1,1\}^n} f(\sigma)\mu_{\beta,\mathbf{X}}(\sigma), \quad \langle f(\sigma)\rangle^{\mathsf{bis}}_{\beta,\mathbf{X}} = \sum_{\sigma\in A_n} f(\sigma)\mu^{\mathsf{bis}}_{\beta,\mathbf{X}}(\sigma).$$

We can extend this definition to functions $f : \{-1,1\}^{n\times k} \to \mathbb{R}$ which take multiple configurations as inputs, in which case we assume that the configurations are sampled independently from the Gibbs distribution:

$$\langle f(\sigma_1,\ldots,\sigma_k)\rangle_{\beta,\mathbf{X}} = \sum_{\sigma_1,\ldots,\sigma_k\in\{-1,1\}^n} \mu_{\beta,\mathbf{X}}(\sigma_1)\cdots\mu_{\beta,\mathbf{X}}(\sigma_k)f(\sigma_1,\ldots,\sigma_k),$$

and the average over μ^{bis} is defined similarly. We sometimes drop the subscripts if they are clear from context. One function of particular interest to us is the *overlap* between two configurations, defined as

$$R_{1,2}(\sigma_1,\sigma_2) = \frac{1}{n}\sum_{i=1}^{n}\sigma_1(i)\sigma_2(i).$$

$R_{1,2}$ gives the normalized inner product between two configurations. If $\langle|R_{1,2}(\sigma_1,\sigma_2)|\rangle_{\beta,\mathbf{X}}$ is close to zero, then two configurations independently sampled from the Gibbs distribution $\mu_{\beta,\mathbf{X}}(\sigma)$ are nearly orthogonal.

Free energy

The free energy for these models is defined as follows.

$$\Phi_{n,\mathrm{SK}}(\beta) = \frac{1}{n}\mathop{\mathbf{E}}_{\mathbf{g}}\left[\log Z(\beta,\mathbf{g})\right], \quad \Phi_{n,d}(\beta) = \frac{1}{n}\mathop{\mathbf{E}}_{\mathbf{A}}\left[\log Z(\beta,\mathbf{A})\right].$$

We can also define free energy in a similar way when restricted to bisections.

$$\Phi^{\mathsf{bis}}_{n,\mathrm{SK}}(\beta) = \frac{1}{n}\mathop{\mathbf{E}}_{\mathbf{g}}\left[\log Z^{\mathsf{bis}}(\beta,\mathbf{g})\right], \quad \Phi^{\mathsf{bis}}_{n,d}(\beta) = \frac{1}{n}\mathop{\mathbf{E}}_{\mathbf{A}}\left[\log Z^{\mathsf{bis}}(\beta,\mathbf{A})\right].$$

The following proposition is a simple consequence of Cauchy-Schwarz.

▶ **Proposition 5.** *Both* $\log Z(\beta,\mathbf{X})$ *and* $\log Z^{\mathsf{bis}}(\beta,\mathbf{X})$ *are convex in* β.

Proof. For any β_1,β_2 we have

$$\begin{aligned} Z\left(\frac{\beta_1+\beta_2}{2},\mathbf{X}\right) &= \sum_{\sigma\in\{-1,1\}^n}\exp\left(-\frac{\beta_1+\beta_2}{2}H(\sigma;\mathbf{X})\right) \\ &\le \sqrt{\left(\sum_{\sigma\in\{-1,1\}^n}\exp\left(-\beta_1 H(\sigma;\mathbf{X})\right)\right)\left(\sum_{\sigma\in\{-1,1\}^n}\exp\left(-\beta_2 H(\sigma;\mathbf{X})\right)\right)} \\ &= \sqrt{Z(\beta_1,\mathbf{X})Z(\beta_2,\mathbf{X})}. \end{aligned}$$

Taking the logarithm of both sides, we get that $\log Z(\beta,\mathbf{X})$ is convex in β. The convexity of $\log Z^{\mathsf{bis}}(\beta,\mathbf{X})$ can be shown in the same way. ◀

By taking expectation over the randomness of $\mathbf{X}$ we immediately obtain the following corollary.

▶ **Corollary 6.** $\Phi_{n,\mathrm{SK}}(\beta), \Phi^{\mathsf{bis}}_{n,\mathrm{SK}}(\beta), \Phi_{n,d}(\beta), \Phi^{\mathsf{bis}}_{n,d}(\beta)$ *are all convex in* β.

Correspondence between dense and sparse models

It is known that as the average degree d gets larger, the sparse model will "converge" to the dense model in some sense. Dembo, Montanari, and Sen proved this for the free energy of these two models using a clever interpolation argument:

▶ **Lemma 7** ([21]). *There exist constants $c_1, c_2, c_3 > 0$ independent of n, β, d such that for any $\beta \in \mathbb{R}$,*

$$\left|\Phi^{\mathsf{bis}}_{n,d}\left(\frac{\beta}{\sqrt{d}}\right) - \Phi^{\mathsf{bis}}_{n,\mathrm{SK}}(\beta)\right| \le c_1\frac{|\beta|^3}{\sqrt{d}} + c_2\frac{\beta^4}{d} + c_3|\beta|\sqrt{d}\cdot\frac{1}{n^2}.^3$$

In this paper, we further extend this correspondence to the average energy $\langle H(\sigma)\rangle$ as well as the average overlap $\langle R_{1,2}(\sigma_1,\sigma_2)^2\rangle$.

▶ **Proposition 8** (Restatement of Proposition 3).
(a) *For every $\beta \ge 0$, we have*

$$\lim_{d\to\infty}\limsup_{n\to\infty}\frac{1}{n}\left|\mathbf{E}_{\mathbf{g}}[\langle H_{\mathrm{SK}}(\sigma)\rangle_{\beta,\mathbf{g}}] - \frac{1}{\sqrt{d}}\mathbf{E}_{\mathbf{A}}[\langle H_d(\sigma)\rangle_{\beta/\sqrt{d},\mathbf{A}}]\right| = 0.$$

(b) *For every $\beta \in \mathbb{R}$, we have*

$$\lim_{d\to\infty}\limsup_{n\to\infty}\frac{1}{n}\left|\mathbf{E}_{\mathbf{g}}[\langle H_{\mathrm{SK}}(\sigma)\rangle^{\mathsf{bis}}_{\beta,\mathbf{g}}] - \frac{1}{\sqrt{d}}\mathbf{E}_{\mathbf{A}}[\langle H_d(\sigma)\rangle^{\mathsf{bis}}_{\beta/\sqrt{d},\mathbf{A}}]\right| = 0.$$

▶ **Proposition 9** (Restatement of Proposition 4).
(a) *For any $\beta \ge 0$,*

$$\lim_{d\to\infty}\limsup_{n\to\infty}\left|\mathbf{E}_{\mathbf{g}}[\langle R_{1,2}^2\rangle_{\beta,\mathbf{g}}] - \mathbf{E}_{\mathbf{A}}[\langle R_{1,2}^2\rangle_{\beta/\sqrt{d},\mathbf{A}}]\right| = 0.$$

(b) *For any $\beta \in \mathbb{R}$,*

$$\lim_{d\to\infty}\limsup_{n\to\infty}\left|\mathbf{E}_{\mathbf{g}}[\langle R_{1,2}^2\rangle^{\mathsf{bis}}_{\beta,\mathbf{g}}] - \mathbf{E}_{\mathbf{A}}[\langle R_{1,2}^2\rangle^{\mathsf{bis}}_{\beta/\sqrt{d},\mathbf{A}}]\right| = 0.$$

We prove Proposition 8 in Section 3 and Proposition 9 in Section 4.

Disorder chaos and stable algorithms

Disorder chaos is a well-studied phenomenon in spin glass theory. The term "disorder" refers to the random matrix $\mathbf{X}$, and "chaos" describes what happens to the Gibbs distribution if we slightly perturb $\mathbf{X}$. For the SK model, consider the following notion of perturbation. Let $\mathbf{g}, \mathbf{g}'$ be two independent copies of Gaussian matrices where $\mathbf{g}_{ij}, \mathbf{g}'_{ij} \sim N(0, 1/2n)$ independently for each i, j. Given any perturbation parameter $t \ge 0$, we can consider the two measures $\mu_{\beta,\mathbf{g}}(\sigma)$ and $\mu_{\beta,\mathbf{g}_t}(\sigma)$ where $\mathbf{g}_t = \sqrt{1-t}\mathbf{g} + \sqrt{t}\mathbf{g}'$. Define $\langle f(\sigma_1,\sigma_2)\rangle_{\beta,\mathbf{g},\mathbf{g}_t}$ to be the average of $f(\sigma_1,\sigma_2)$ where $\sigma_1 \sim \mu_{\beta,\mathbf{g}}$ and $\sigma_2 \sim \mu_{\beta,\mathbf{g}_t}$ sampled independently from these two distributions respectively, i.e.,

$$\langle f(\sigma_1,\sigma_2)\rangle_{\beta,\mathbf{g},\mathbf{g}_t} = \sum_{\sigma_1,\sigma_2\in\{-1,1\}^n} f(\sigma_1,\sigma_2)\mu_{\beta,\mathbf{g}}(\sigma_1)\mu_{\beta,\mathbf{g}_t}(\sigma_2).$$

It is known that two such samples σ_1, σ_2 will likely have some nontrivial overlap when $t = 0$, in which case there is no perturbation and $\langle f(\sigma_1,\sigma_2)\rangle_{\beta,\mathbf{g},\mathbf{g}_t}$ is simply $\langle f(\sigma_1,\sigma_2)\rangle_{\beta,\mathbf{g}}$.

▶ **Theorem 10** (See e.g. [4]). *If $|\beta| > 1$, then there exists $\epsilon = \epsilon(\beta) > 0$ such that*

$$\lim_{n\to\infty} \mathbf{E}_{\mathbf{g}} \left[\langle R_{1,2}(\sigma_1,\sigma_2)^2\rangle_{\beta,\mathbf{g}}\right] \geq \epsilon(\beta). \tag{2.1}$$

However, when $t > 0$, then the overlap will be zero in the $n \to \infty$ limit.

▶ **Theorem 11** ([12]). *For all $\beta \in \mathbb{R}, t \in (0,1]$, we have*

$$\lim_{n\to\infty} \mathbf{E}_{\mathbf{g},\mathbf{g}'} \left[\langle R_{1,2}(\sigma_1,\sigma_2)^2\rangle_{\beta,\mathbf{g},\mathbf{g}_t}\right] = 0. \tag{2.2}$$

If we think of the expression on the left hand side of (2.1) and (2.2) as a function of t, then (2.1) and (2.2) together suggest that this function is not right-continuous at $t = 0$ if $|\beta| > 1$. It was shown in [4] that this property very naturally obstructs a class of sampling algorithms for $\mu_{\beta,\mathbf{g}}$.

▶ **Definition 12** (Definition 2.2 in [4]). *Let $\{\mathsf{ALG}_n\}_{n\geq 1}$ be a family of sampling algorithms for the Gibbs measure $\mu_{\beta,\mathbf{g}}$, where ALG_n takes an $n \times n$ matrix $\mathbf{g}$ and $\beta \in \mathbb{R}$ as input and outputs an assignment in $\{-1,1\}^n$. Let $\mu^{\mathsf{ALG}}_{\beta,\mathbf{g}}$ be the output law of ALG_n. We say that $\{\mathsf{ALG}_n\}_{n\geq 1}$ is stable (with respect to disorder at β), if*

$$\lim_{t\to 0} \limsup_{n\to\infty} \mathbf{E}\left[W_{2,n}(\mu^{\mathsf{ALG}}_{\beta,\mathbf{g}}, \mu^{\mathsf{ALG}}_{\beta,\mathbf{g}_t})\right] = 0. \tag{2.3}$$

Intuitively, (2.3) means that stable algorithms are not able to make the leap at the discontinuity $t = 0$ implied by (2.1) and (2.2), and therefore must be producing a distribution that is bounded away from the Gibbs distribution $\mu_{\beta,\mathbf{g}}$ in terms of the $W_{2,n}$ distance. This intuition is formalized in [4] as the following theorem.

▶ **Theorem 13** (Theorem 2.6 in [4]). *Fix β such that $|\beta| > 1$. Let $\{\mathsf{ALG}_n\}_{n\geq 1}$ be a family of sampling algorithms for the Gibbs measure $\mu_{\beta,\mathbf{g}}$ that is stable with respect to disorder at β, then*

$$\liminf_{n\to\infty} \mathbf{E}_{\mathbf{g}} \left[W_{2,n}(\mu_{\beta,\mathbf{g}}, \mu^{\mathsf{ALG}}_{\beta,\mathbf{g}})\right] > 0.$$

In this paper, we obstruct stable sampling algorithms for sparse models as well. For the sparse models, we consider a slightly different notion of perturbation. Fix the average degree $d > 0$ and the perturbation parameter $t \in [0,1]$, we will take three independent random matrices $\mathbf{A}^{(1-t)}, \mathbf{A}^{(t,1)}, \mathbf{A}^{(t,2)}$ such that $\mathbf{A}^{(1-t)}_{ij} \sim \mathrm{Po}((1-t)d/(2n))$, $\mathbf{A}^{(t,1)}_{ij}, \mathbf{A}^{(t,2)}_{ij} \sim \mathrm{Po}(td/(2n))$ independently for all $i, j \in [n]$. We then define $\mathbf{A} = \mathbf{A}^{(1-t)} + \mathbf{A}^{(t,1)}$ and $\mathbf{A}_t = \mathbf{A}^{(1-t)} + \mathbf{A}^{(t,2)}$. Similarly to Definition 12, we say that a family of sampling algorithms $\{\mathsf{ALG}_n\}_{n\geq 1}$ with inputs β and $\mathbf{A}$ is stable at inverse temperature β if

$$\lim_{t\to 0} \limsup_{n\to\infty} \mathbf{E}\left[W_{2,n}(\mu^{\mathsf{ALG}}_{\beta,\mathbf{A}}, \mu^{\mathsf{ALG}}_{\beta,\mathbf{A}_t})\right] = 0. \tag{2.4}$$

As an example, it was recently shown that stable algorithms implemented by Boolean circuits with bounded depth are stable [1].

When $t = 0$, we have by Proposition 9 and Theorem 10 that for $\beta > 1$,

$$\lim_{d\to\infty} \liminf_{n\to\infty} \mathbf{E}_{\mathbf{A}}[\langle R^2_{1,2}\rangle_{\beta/\sqrt{d},\mathbf{A}}] > 0, \tag{2.5}$$

and for β with $|\beta| > 1$

$$\lim_{d\to\infty} \liminf_{n\to\infty} \mathbf{E}_{\mathbf{A}}[\langle R^2_{1,2}\rangle^{\mathsf{bis}}_{\beta/\sqrt{d},\mathbf{A}}] > 0.$$

On the other hand, if $t > 0$, we show in the following lemma that the overlap tends to zero as n goes to infinity.

▶ **Lemma 14.** *Fix $t > 0$.*

(a) *For any $\beta \geq 0$,*

$$\lim_{d\to\infty} \limsup_{n\to\infty} \mathbf{E}[\langle R_{1,2}^2 \rangle_{\beta/\sqrt{d},\mathbf{A},\mathbf{A}_t}] = 0. \tag{2.6}$$

(b) *For any $\beta \in \mathbb{R}$,*

$$\lim_{d\to\infty} \limsup_{n\to\infty} \mathbf{E}[\langle R_{1,2}^2 \rangle^{\mathsf{bis}}_{\beta/\sqrt{d},\mathbf{A},\mathbf{A}_t}] = 0.$$

The following lemma establishes the connection between the overlap and the $W_{2,n}$ distance.

▶ **Lemma 15** (Lemma 5.3 in [4]). *Fix $n \in \mathbb{N}$. Let $\mu_1, \mu_2, \nu_1, \nu_2$ be distributions over $\{-1,+1\}^n$. We have*

$$\left| \mathop{\mathbf{E}}_{(\sigma,\sigma')\sim\mu_1\otimes\nu_1} [|R_{1,2}(\sigma,\sigma')|] - \mathop{\mathbf{E}}_{(\sigma,\sigma')\sim\mu_2\otimes\nu_2} [|R_{1,2}(\sigma,\sigma')|] \right| \leq W_{2,n}(\mu_1,\mu_2) + W_{2,n}(\nu_1,\nu_2).$$

We now have the ingredients needed for proving Theorem 1. Theorem 2 can be proved in the same way and we omit the details here.

Proof of Theorem 1. We drop the inverse temperature parameter in the $\langle\cdot\rangle$ notation for brevity. For any arbitrary $t > 0$, by choosing $\mu_1 = \mu_2 = \nu_1 = \mu_{\beta/\sqrt{d},\mathbf{A}}$ and $\nu_2 = \mu_{\beta/\sqrt{d},\mathbf{A}_t}$ in Lemma 15, we get that

$$W_{2,n}(\mu_{\beta/\sqrt{d},\mathbf{A}}, \mu_{\beta/\sqrt{d},\mathbf{A}_t}) \geq \left| \langle |R_{1,2}(\sigma,\sigma')| \rangle_{\mathbf{A}} - \langle |R_{1,2}(\sigma,\sigma')| \rangle_{\mathbf{A},\mathbf{A}_t} \right|.$$

Taking expectation on both sides, we obtain

$$\begin{aligned} \mathop{\mathbf{E}}_{\mathbf{A},\mathbf{A}_t} \left[W_{2,n}(\mu_{\beta/\sqrt{d},\mathbf{A}}, \mu_{\beta/\sqrt{d},\mathbf{A}_t}) \right] &\geq \mathop{\mathbf{E}}_{\mathbf{A},\mathbf{A}_t} \left[\left| \langle |R_{1,2}(\sigma,\sigma')| \rangle_{\mathbf{A}} - \langle |R_{1,2}(\sigma,\sigma')| \rangle_{\mathbf{A},\mathbf{A}_t} \right| \right] \\ &\geq \mathop{\mathbf{E}}_{\mathbf{A},\mathbf{A}_t} \left[\langle |R_{1,2}(\sigma,\sigma')| \rangle_{\mathbf{A}} \right] - \mathop{\mathbf{E}}_{\mathbf{A},\mathbf{A}_t} \left[\langle |R_{1,2}(\sigma,\sigma')| \rangle_{\mathbf{A},\mathbf{A}_t} \right]. \end{aligned}$$

In the second inequality we used the fact that $|a-b| \geq |a| - |b|$ for any $a, b \in \mathbb{R}$. Taking lim inf on both sides, we obtain

$$\begin{aligned} &\liminf_{n\to\infty} \mathop{\mathbf{E}}_{\mathbf{A},\mathbf{A}_t} \left[W_{2,n}(\mu_{\beta/\sqrt{d},\mathbf{A}}, \mu_{\beta/\sqrt{d},\mathbf{A}_t}) \right] \\ \geq &\liminf_{n\to\infty} \mathop{\mathbf{E}}_{\mathbf{A},\mathbf{A}_t} \left[\langle |R_{1,2}(\sigma,\sigma')| \rangle_{\mathbf{A}} \right] - \limsup_{n\to\infty} \mathop{\mathbf{E}}_{\mathbf{A},\mathbf{A}_t} \left[\langle |R_{1,2}(\sigma,\sigma')| \rangle_{\mathbf{A},\mathbf{A}_t} \right]. \end{aligned}$$

By (2.5) and (2.6), if d is sufficiently large, for some $\epsilon = \epsilon(\beta) > 0$ independent of t,

$$\liminf_{n\to\infty} \mathop{\mathbf{E}}_{\mathbf{A},\mathbf{A}_t} \left[W_{2,n}(\mu_{\beta/\sqrt{d},\mathbf{A}}, \mu_{\beta/\sqrt{d},\mathbf{A}_t}) \right] \geq \epsilon.$$

By the triangle inequality, we have

$$\begin{aligned} &W_{2,n}(\mu_{\beta/\sqrt{d},\mathbf{A}}, \mu_{\beta/\sqrt{d},\mathbf{A}_t}) \\ \leq &W_{2,n}(\mu_{\beta/\sqrt{d},\mathbf{A}}, \mu^{\mathsf{ALG}}_{\beta/\sqrt{d},\mathbf{A}}) + W_{2,n}(\mu^{\mathsf{ALG}}_{\beta/\sqrt{d},\mathbf{A}}, \mu^{\mathsf{ALG}}_{\beta/\sqrt{d},\mathbf{A}_t}) + W_{2,n}(\mu_{\beta/\sqrt{d},\mathbf{A}_t}, \mu^{\mathsf{ALG}}_{\beta/\sqrt{d},\mathbf{A}_t}). \end{aligned}$$

Since $\mathbf{A}$ and $\mathbf{A}_t$ have the same distribution, we have

$$\mathbf{E}\left[W_{2,n}(\mu_{\beta/\sqrt{d},\mathbf{A}}, \mu^{\mathsf{ALG}}_{\beta/\sqrt{d},\mathbf{A}}) \right] = \mathbf{E}\left[W_{2,n}(\mu_{\beta/\sqrt{d},\mathbf{A}_t}, \mu^{\mathsf{ALG}}_{\beta/\sqrt{d},\mathbf{A}_t}) \right].$$

Since $\{\mathsf{ALG}_n\}_n$ is stable at any inverse temperature, by (2.4) and taking t sufficiently small we have

$$\limsup_{n\to\infty} \mathbf{E}\left[W_{2,n}(\mu^{\mathsf{ALG}}_{\beta/\sqrt{d},\mathbf{A}}, \mu^{\mathsf{ALG}}_{\beta/\sqrt{d},\mathbf{A}_t})\right] \le \frac{\epsilon}{2}.$$

It follows that

$$\liminf_{n\to\infty} \mathbf{E}\left[W_{2,n}(\mu^{\mathsf{ALG}}_{\beta/\sqrt{d},\mathbf{A}}, \mu_{\beta/\sqrt{d},\mathbf{A}})\right] \ge \frac{\epsilon}{4}. \qquad \blacktriangleleft$$

Free energy of coupled bisection models

In subsequent proofs, we will need the following results on the free energy for coupled bisection models. More specifically, we show that the limiting free energy does not change under the bisection constraint for both dense and sparse models (for the sparse model we require $d \to \infty$). Intuitively, in the case of SK model this holds because the symmetry of its disorder implies that there is nothing special about the all-one direction. The sparse case requires more care since it no longer has the symmetry, but we overcome this difficulty using Poisson concentration arguments. Due to the space constraint, proofs will be omitted here and we refer interested readers to the full version of this paper.

Let us define (β is omitted from the notation for simplicity)

$$Z^S_{\mathbf{g},\mathbf{g}_t} = \sum_{\substack{\sigma_1,\sigma_2\in\{-1,1\}^n \\ R_{1,2}(\sigma_1,\sigma_2)\in S}} \exp\left(-\beta(H(\sigma_1;\mathbf{g}) + H(\sigma_2;\mathbf{g}_t))\right)$$

and

$$Z^{S,\mathsf{bis}}_{\mathbf{g},\mathbf{g}_t} = \sum_{\substack{\sigma_1,\sigma_2\in A_n \\ R_{1,2}(\sigma_1,\sigma_2)\in S}} \exp\left(-\beta(H(\sigma_1;\mathbf{g}) + H(\sigma_2;\mathbf{g}_t))\right).$$

Similarly for the sparse model, let us define

$$Z^S_{\mathbf{A},\mathbf{A}_t} = \sum_{\substack{\sigma_1,\sigma_2\in\{-1,1\}^n \\ R_{1,2}(\sigma_1,\sigma_2)\in S}} \exp\left(-\frac{\beta}{\sqrt{d}}\cdot(H(\sigma_1;\mathbf{A}) + H(\sigma_2;\mathbf{A}_t))\right),$$

$$Z^{S,\mathsf{bis}}_{\mathbf{A},\mathbf{A}_t} = \sum_{\substack{\sigma_1,\sigma_2\in A_n \\ R_{1,2}(\sigma_1,\sigma_2)\in S}} \exp\left(-\frac{\beta}{\sqrt{d}}\cdot(H(\sigma_1;\mathbf{A}) + H(\sigma_2;\mathbf{A}_t))\right).$$

▶ **Theorem 16.** *Let $S \subseteq [-1,1]$. For any $\beta \in \mathbb{R}$, we have*

$$\lim_{n\to\infty} \frac{1}{n}\mathbf{E}\left[\log Z^{S,\mathsf{bis}}_{\mathbf{g},\mathbf{g}_t} - \log Z^S_{\mathbf{g},\mathbf{g}_t}\right] = 0.$$

By letting $t = 0$ and $S = [-1,1]$, we immediately obtain the following.

▶ **Corollary 17.** *We have*

$$\lim_{n\to\infty} \left(\Phi^{\mathsf{bis}}_{n,\mathrm{SK}}(\beta) - \Phi_{n,\mathrm{SK}}(\beta)\right) = 0.$$

▶ **Lemma 18.** *Fix $\beta, d > 0$ and $t \in [0,1]$. For every $\delta \in (0,1/2)$, there exists a constant $C = C(\beta,d)$ such that for all sufficiently large n*

$$\Pr\left[\left|\log Z^S_{\mathbf{A},\mathbf{A}_t} - \mathbf{E}[\log Z^S_{\mathbf{A},\mathbf{A}_t}]\right| \ge n^{1/2+\delta}\right] \le \exp\left(-C\cdot n^{2\delta}\right)$$

and

$$\Pr\left[\left|\log Z^{S,\mathsf{bis}}_{\mathbf{A},\mathbf{A}_t} - \mathbf{E}[\log Z^{S,\mathsf{bis}}_{\mathbf{A},\mathbf{A}_t}]\right| \ge n^{1/2+\delta}\right] \le \exp\left(-C\cdot n^{2\delta}\right).$$

▶ **Theorem 19.** *Let $0 \le a < b$. For every $\epsilon > 0$, if d is sufficiently large, then for every $\beta \in [a, b]$,*

$$\limsup_{n\to\infty} \frac{1}{n} \left| \mathbf{E} \log Z^{S}_{\mathbf{A},\mathbf{A}_t} - \mathbf{E} \log Z^{S,\mathsf{bis}}_{\mathbf{A},\mathbf{A}_t} \right| \le \epsilon.$$

Again, by letting $t = 0$ and $S = [-1, 1]$, we immediately obtain the following.

▶ **Theorem 20.** *Let $0 \le a < b$. For every $\epsilon > 0$, if d is sufficiently large, then for every $\beta \in [a, b]$,*

$$\limsup_{n\to\infty} \left| \Phi_{n,d}\left(\frac{\beta}{\sqrt{d}}\right) - \Phi^{\mathsf{bis}}_{n,d}\left(\frac{\beta}{\sqrt{d}}\right) \right| \le \epsilon.$$

3 Interpolating the average energy between SK and the diluted model

In this section we prove Proposition 8 and establish the interpolation of average energy between sparse and dense models. By Lemma 7, we already know that the free energy of the sparse model converges to that of the dense model as the average degree d goes to infinity. The following connection between free energy and average energy is well-known:

$$\frac{\partial}{\partial\beta} \mathbf{E}\left[\log Z(\beta, \mathbf{X})\right] = \mathbf{E}\left[\frac{\sum_{\sigma\in\{-1,1\}^n} -H(\sigma;\mathbf{X}) \exp(-\beta H(\sigma;\mathbf{X}))}{Z(\beta,\mathbf{X})}\right] = -\mathbf{E}\left[\langle H(\sigma;\mathbf{X})\rangle_{\beta,\mathbf{X}}\right]. \tag{3.1}$$

Similarly,

$$\frac{\partial}{\partial\beta} \mathbf{E}\left[\log Z^{\mathsf{bis}}(\beta, \mathbf{X})\right] = \mathbf{E}\left[\frac{\sum_{\sigma\in A_n} -H(\sigma;\mathbf{X}) \exp(-\beta H(\sigma;\mathbf{X}))}{Z^{\mathsf{bis}}(\beta,\mathbf{X})}\right] = -\mathbf{E}\left[\langle H(\sigma;\mathbf{X})\rangle^{\mathsf{bis}}_{\beta,\mathbf{X}}\right]. \tag{3.2}$$

To obtain convergence of the partial derivatives, we use the following elementary fact sometimes known as Griffith's lemma in statistical physics.

▶ **Proposition 21** (See e.g. [41])**.** *Let $(f_\alpha)_{\alpha\ge 0}$ be a family of convex and differentiable functions that converges pointwise in an open interval I to f, then $\lim_{\alpha\to\infty} f'_\alpha(x) = f'(x)$ at every $x \in I$ where $f'(x)$ exists.*

Proof of Proposition 8. Proposition 8 says that

(a) For every $\beta \ge 0$, $\lim_{d\to\infty} \limsup_{n\to\infty} \frac{1}{n} \left| \mathbf{E}_{\mathbf{g}}[\langle H_{\mathrm{SK}}(\sigma)\rangle_{\beta,\mathbf{g}}] - \frac{1}{\sqrt{d}} \mathbf{E}_{\mathbf{A}}[\langle H_d(\sigma)\rangle_{\beta/\sqrt{d},\mathbf{A}}] \right| = 0.$

(b) For every $\beta \in \mathbb{R}$, $\lim_{d\to\infty} \limsup_{n\to\infty} \frac{1}{n} \left| \mathbf{E}_{\mathbf{g}}[\langle H_{\mathrm{SK}}(\sigma)\rangle^{\mathsf{bis}}_{\beta,\mathbf{g}}] - \frac{1}{\sqrt{d}} \mathbf{E}_{\mathbf{A}}[\langle H_d(\sigma)\rangle^{\mathsf{bis}}_{\beta/\sqrt{d},\mathbf{A}}] \right| = 0.$

We first prove part (b). Assume for the sake of contradiction that for some $\beta_0 \in \mathbb{R}$

$$\limsup_{d\to\infty} \limsup_{n\to\infty} \frac{1}{n} \left| \mathbf{E}_{\mathbf{g}}[\langle H_{\mathrm{SK}}(\sigma)\rangle^{\mathsf{bis}}_{\beta_0,\mathbf{g}}] - \frac{1}{\sqrt{d}} \mathbf{E}_{\mathbf{A}}[\langle H_d(\sigma)\rangle^{\mathsf{bis}}_{\beta_0/\sqrt{d},\mathbf{A}}] \right| = \epsilon > 0.$$

It follows that we can choose a sequence of pairs $(d_i, n_i)_{i\in\mathbb{N}}$ such that

$$\lim_{i\to\infty} \frac{1}{n_i} \left| \mathbf{E}_{\mathbf{g}}[\langle H_{\mathrm{SK}}(\sigma)\rangle^{\mathsf{bis}}_{\beta_0,\mathbf{g}}] - \frac{1}{\sqrt{d_i}} \mathbf{E}_{\mathbf{A}}[\langle H_{d_i}(\sigma)\rangle^{\mathsf{bis}}_{\beta_0/\sqrt{d_i},\mathbf{A}}] \right| = \epsilon. \tag{3.3}$$

Note that for every i, n_i can be chosen to be sufficiently large so that the $o_n(1)$ term in Lemma 7 is $\leq 1/d_i$, and under this assumption we have $\left(\Phi^{\mathsf{bis}}_{n_i,d_i}\left(\frac{\beta}{\sqrt{d_i}}\right)\right)_{i\in\mathbb{N}}$ as a family of functions of β converges pointwise[4] to $\lim_{i\to\infty}\Phi^{\mathsf{bis}}_{n_i,\mathrm{SK}}(\beta)$. Here we remark that it is known that the limit $\lim_{n\to\infty}\Phi_{n,\mathrm{SK}}(\beta)$ exists and is differentiable in β [39, 40], and by Corollary 17 we have

$$\lim_{n\to\infty}\Phi_{n,\mathrm{SK}}(\beta)=\lim_{n\to\infty}\Phi^{\mathsf{bis}}_{n,\mathrm{SK}}(\beta)=\lim_{i\to\infty}\Phi^{\mathsf{bis}}_{n_i,\mathrm{SK}}(\beta).$$

By Proposition 21, we have

$$\lim_{i\to\infty}\frac{\partial}{\partial\beta}\Phi^{\mathsf{bis}}_{n_i,d_i}\left(\frac{\beta}{\sqrt{d_i}}\right)=\frac{\partial}{\partial\beta}\lim_{n\to\infty}\Phi^{\mathsf{bis}}_{n,\mathrm{SK}}(\beta)=\lim_{n\to\infty}\frac{\partial}{\partial\beta}\Phi^{\mathsf{bis}}_{n,\mathrm{SK}}(\beta)=\lim_{i\to\infty}\frac{\partial}{\partial\beta}\Phi^{\mathsf{bis}}_{n_i,\mathrm{SK}}(\beta) \quad (3.4)$$

for every $\beta\in\mathbb{R}$. By (3.2), we have

$$\frac{\partial}{\partial\beta}\Phi^{\mathsf{bis}}_{n,d}\left(\frac{\beta}{\sqrt{d}}\right)=-\frac{1}{\sqrt{d}\cdot n}\mathop{\mathbf{E}}_{\mathbf{A}}\left[\langle H_d(\sigma)\rangle^{\mathsf{bis}}_{\beta/\sqrt{d},\mathbf{A}}\right], \quad (3.5)$$

and

$$\frac{\partial}{\partial\beta}\Phi^{\mathsf{bis}}_{n,\mathrm{SK}}(\beta)=-\frac{1}{n}\mathop{\mathbf{E}}_{\mathbf{g}}[\langle H_{\mathrm{SK}}(\sigma)\rangle^{\mathsf{bis}}_{\beta,\mathbf{g}}]. \quad (3.6)$$

(3.4), (3.5), and (3.6) imply that

$$\lim_{i\to\infty}\frac{1}{n_i}\left|\mathop{\mathbf{E}}_{\mathbf{g}}[\langle H_{\mathrm{SK}}(\sigma)\rangle^{\mathsf{bis}}_{\beta_0,\mathbf{g}}]-\frac{1}{\sqrt{d_i}}\mathop{\mathbf{E}}_{\mathbf{A}}[\langle H_{d_i}(\sigma)\rangle^{\mathsf{bis}}_{\beta_0/\sqrt{d_i},\mathbf{A}}]\right|=0.$$

However, this contradicts (3.3), so we obtain part (b).

For part (a), note that by Theorem 20, for any bounded open interval $I\subset[0,+\infty)$, we can choose the sequence $(d_i,n_i)_{i\in\mathbb{N}}$ to satisfy the additional requirement that

$$\lim_{i\to\infty}\Phi^{\mathsf{bis}}_{n_i,d_i}(\beta/\sqrt{d_i})=\lim_{i\to\infty}\Phi_{n_i,d_i}(\beta/\sqrt{d_i})$$

for all $\beta\in I$. In particular, we can choose I to include any desired $\beta_0>0$ (when $\beta=0$ the Gibbs measure is uniform and part (a) trivially holds). We can then invoke Proposition 21 again and proceed with the same proof. ◀

4 Interpolating the average overlap between SK and the diluted model

In this section we prove Proposition 9. Since the proofs are mostly the same, we will only prove item (a) in Proposition 9. We first state the following well-known fact which relates the average energy to the average overlap of the Gibbs distribution in the SK model.

▶ **Lemma 22** (See e.g. [32]). *For every $\beta\in\mathbb{R}$, we have*

$$\frac{1}{n}\mathop{\mathbf{E}}_{\mathbf{g}}\left[\langle H_{\mathrm{SK}}(\sigma)\rangle\right]=\frac{\beta}{2}\mathop{\mathbf{E}}_{\mathbf{g}}[1-\langle R^2_{1,2}\rangle_{\beta,\mathbf{g}}].$$

The proof of Lemma 22 uses the following proposition.

[4] The convergence is in fact uniform in any bounded interval.

▶ **Proposition 23.** *Let $\mathbf{X} \in \mathbb{R}^{n\times n}, \beta \in \mathbb{R}$. We have*

$$\sum_{\sigma\in\{-1,1\}^n} \sum_{i,j=1}^{n} \sigma(i)\sigma(j)\frac{\partial}{\partial \mathbf{X}_{ij}}\mu(\sigma;\beta,\mathbf{X}) = -\beta n^2\left(1-\langle R_{1,2}^2\rangle_{\beta,\mathbf{X}}\right)$$

Proof. For every $\sigma \in \{-1,1\}^n$ and $i,j \in [n]$, we have

$$\begin{aligned}
&\frac{\partial}{\partial \mathbf{X}_{ij}}\mu(\sigma;\beta,\mathbf{X}) \\
=&\frac{\partial}{\partial \mathbf{X}_{ij}}\left(\frac{\exp\left(-\beta H(\sigma;\mathbf{X})\right)}{Z(\beta,\mathbf{X})}\right) \\
=&(-\beta)\left(\sigma(i)\sigma(j)\mu(\sigma;\beta,\mathbf{X}) - \sum_{\sigma'\in\{-1,1\}^n}\sigma'(i)\sigma'(j)\cdot\mu(\sigma;\beta,\mathbf{X})\mu(\sigma';\beta,\mathbf{X})\right). \qquad (4.1)
\end{aligned}$$

Summing over all σ, i, j, we obtain

$$\begin{aligned}
&\sum_{\sigma\in\{-1,1\}^n} \sum_{i,j=1}^{n} \sigma(i)\sigma(j)\frac{\partial}{\partial \mathbf{X}_{ij}}\mu(\sigma;\beta,\mathbf{X}) \\
=&\sum_{\sigma\in\{-1,1\}^n} \sum_{i,j=1}^{n} \sigma(i)\sigma(j)\cdot(-\beta)\Bigg(\sigma(i)\sigma(j)\mu(\sigma;\beta,\mathbf{X}) \\
&\qquad\qquad - \sum_{\sigma'\in\{-1,1\}^n}\sigma'(i)\sigma'(j)\cdot\mu(\sigma;\beta,\mathbf{X})\mu(\sigma';\beta,\mathbf{X})\Bigg) \\
=&-\beta n^2\left(1-\langle R_{1,2}^2\rangle_{\beta,\mathbf{X}}\right). \qquad \blacktriangleleft
\end{aligned}$$

Lemma 22 can be proved using Proposition 23 and the Gaussian integration by parts formula. We prove a statement similar to Lemma 22, but for the sparse model. We will use the following lemma, known as the Stein-Chen identity for the Poisson distribution, in place of Gaussian integration by parts.

▶ **Lemma 24** ([14])**.** *Let $X \sim \mathrm{Po}(\lambda)$. For any bounded function f, we have*

$$\mathbf{E}[X\cdot f(X)] = \lambda\,\mathbf{E}[f(X+1)].$$

▶ **Lemma 25.** *Let $\mathbf{J}_{ij}$ be the matrix with 1 in the (i,j)-th entry and 0 everywhere else. Then there exists $\xi \in [0,1]$ such that*

$$\left|\mu(\sigma;\beta/\sqrt{d},\mathbf{X}+\mathbf{J}_{ij}) - \mu(\sigma;\beta/\sqrt{d},\mathbf{X}) - \frac{\partial}{\partial \mathbf{X}_{ij}}\mu(\sigma;\beta/\sqrt{d},\mathbf{X})\right| \le \frac{3\beta^2}{d}\cdot\mu(\sigma;\beta/\sqrt{d},\mathbf{X}+\xi\mathbf{J}_{ij}). \qquad (4.2)$$

Proof. Since μ is infinitely differentiable in each entry of $\mathbf{X}$, by Taylor's Theorem, there exists $\xi \in [0,1]$ such that

$$\mu(\sigma;\beta/\sqrt{d},\mathbf{X}+\mathbf{J}_{ij}) - \mu(\sigma;\beta/\sqrt{d},\mathbf{X}) - \frac{\partial}{\partial \mathbf{X}_{ij}}\mu(\sigma;\beta/\sqrt{d},\mathbf{X}) = \frac{1}{2}\cdot\frac{\partial^2}{\partial \mathbf{X}_{ij}^2}\mu(\sigma;\beta/\sqrt{d},\mathbf{X}+\xi\mathbf{J}_{ij}). \qquad (4.3)$$

It remains to bound $\frac{\partial^2}{\partial \mathbf{X}_{ij}^2}\mu(\sigma;\beta/\sqrt{d},\mathbf{X}+\xi\mathbf{J}_{ij})$. By (4.1), we have (omitting $\beta/\sqrt{d},\mathbf{X}+\xi\mathbf{J}_{ij}$ from the notation for simplicity)

$$\begin{aligned}\frac{\partial^2}{\partial \mathbf{X}_{ij}^2}\mu(\sigma) &= \frac{\partial}{\partial \mathbf{X}_{ij}}\left(\left(-\frac{\beta}{\sqrt{d}}\right)\left(\sigma(i)\sigma(j)-\sum_{\sigma'\in\{-1,1\}^n}\sigma'(i)\sigma'(j)\mu(\sigma')\right)\mu(\sigma)\right)\\ &= \left(-\frac{\beta}{\sqrt{d}}\right)^2\left(-1+\sum_{\sigma',\sigma''\in\{-1,1\}^n}\sigma'(i)\sigma'(j)\sigma''(i)\sigma''(j)\mu(\sigma')\mu(\sigma'')\right)\mu(\sigma)\\ &\quad+\left(-\frac{\beta}{\sqrt{d}}\right)^2\left(\sigma(i)\sigma(j)-\sum_{\sigma'\in\{-1,1\}^n}\sigma'(i)\sigma'(j)\mu(\sigma')\right)^2\mu(\sigma).\end{aligned}$$

Since

$$\left|\sum_{\sigma',\sigma''\in\{-1,1\}^n}\sigma'(i)\sigma'(j)\sigma''(i)\sigma''(j)\mu(\sigma')\mu(\sigma'')\right| = |\langle\sigma'(i)\sigma'(j)\sigma''(i)\sigma''(j)\rangle| \le 1$$

and similarly

$$\left|\sum_{\sigma'\in\{-1,1\}^n}\sigma'(i)\sigma'(j)\mu(\sigma')\right| = |\langle\sigma'(i)\sigma'(j)\rangle| \le 1,$$

it follows that

$$\left|\frac{\partial^2}{\partial \mathbf{X}_{ij}^2}\mu(\sigma)\right| \le \frac{\beta^2}{d}(2+4)\mu(\sigma) = \frac{6\beta^2}{d}\mu(\sigma). \tag{4.4}$$

(4.3) and (4.4) together imply (4.2). ◀

▶ **Proposition 26.** *Let $\mathbf{J}_{ij}$ be the matrix with 1 in the (i,j)-th entry and 0 everywhere else. For any $\xi\in\mathbb{R}$, we have*

$$\mathrm{e}^{-2\beta|\xi|}\cdot\mu(\sigma;\beta,\mathbf{X}) \le \mu(\sigma;\beta,\mathbf{X}+\xi\mathbf{J}_{ij}) \le \mathrm{e}^{2\beta|\xi|}\cdot\mu(\sigma;\beta,\mathbf{X})$$

Proof. We have

$$\exp(-\beta H(\sigma;\mathbf{X}+\xi\mathbf{J}_{ij})) = \exp(-\beta H(\sigma;\mathbf{X})-\beta H(\sigma;\xi\mathbf{J}_{ij})) = \exp(-\beta H(\sigma;\mathbf{X}))\cdot\mathrm{e}^{-\beta\xi\sigma(i)\sigma(j)},$$

which implies that

$$\mathrm{e}^{-\beta|\xi|}\exp(-\beta H(\sigma;\mathbf{X})) \le \exp(-\beta H(\sigma;\mathbf{X}+\xi\mathbf{J}_{ij})) \le \mathrm{e}^{\beta|\xi|}\exp(-\beta H(\sigma;\mathbf{X})).$$

Summing over $\sigma\in\{-1,1\}^n$, we obtain

$$\mathrm{e}^{-\beta|\xi|}Z(\beta,\mathbf{X}) \le Z(\beta,\mathbf{X}+\xi\mathbf{J}_{ij}) \le \mathrm{e}^{\beta|\xi|}Z(\beta,\mathbf{X}).$$

Since these quantities are all strictly positive, we have

$$\mu(\sigma;\beta,\mathbf{X}+\xi\mathbf{J}_{ij}) = \frac{\exp(-\beta H(\sigma;\mathbf{X}+\xi\mathbf{J}_{ij}))}{Z(\beta,\mathbf{X}+\xi\mathbf{J}_{ij})} \le \frac{\mathrm{e}^{\beta|\xi|}\exp(-\beta H(\sigma;\mathbf{X}))}{\mathrm{e}^{-\beta|\xi|}Z(\beta,\mathbf{X})} = \mathrm{e}^{2\beta|\xi|}\cdot\mu(\sigma;\beta,\mathbf{X})$$

and similarly

$$\mathrm{e}^{-2\beta|\xi|}\cdot\mu(\sigma;\beta,\mathbf{X}) \le \mu(\sigma;\beta,\mathbf{X}+\xi\mathbf{J}_{ij}).$$ ◀

▶ **Lemma 27.** *For every $\beta \in \mathbb{R}$, we have*

$$\frac{1}{\sqrt{d}n}\,\mathbf{E}\left[\langle H_d(\sigma)\rangle_{\frac{\beta}{\sqrt{d}},\mathbf{A}}\right] = \frac{\sqrt{d}}{2}\cdot\mathbf{E}\left[\left\langle\left(\frac{1}{n}\sum_{i=1}^{n}\sigma(i)\right)^2\right\rangle_{\frac{\beta}{\sqrt{d}},\mathbf{A}}\right] - \frac{\beta}{2}\,\mathbf{E}\left[1-\langle R_{1,2}^2\rangle_{\frac{\beta}{\sqrt{d}},\mathbf{A}}\right] + O_d\left(\frac{1}{\sqrt{d}}\right).$$

Proof. For simplicity, we drop the subscripts from $\langle\cdot\rangle$. We have

$$\begin{aligned}\frac{1}{\sqrt{d}n}\,\mathbf{E}\left[\langle H_d(\sigma)\rangle\right] &= \frac{1}{\sqrt{d}n}\,\mathbf{E}\left[\sum_{\sigma\in\{-1,1\}^n}\sum_{i,j=1}^{n}\sigma(i)\sigma(j)\mathbf{A}_{ij}\cdot\mu\left(\sigma;\beta/\sqrt{d},\mathbf{A}\right)\right]\\ &= \frac{1}{\sqrt{d}n}\cdot\frac{d}{2n}\cdot\mathbf{E}\left[\sum_{\sigma\in\{-1,1\}^n}\sum_{i,j=1}^{n}\sigma(i)\sigma(j)\mu\left(\sigma;\beta/\sqrt{d},\mathbf{A}+\mathbf{J}_{ij}\right)\right]\\ &= \frac{\sqrt{d}}{2n^2}\cdot\mathbf{E}\left[\sum_{\sigma\in\{-1,1\}^n}\sum_{i,j=1}^{n}\sigma(i)\sigma(j)\mu\left(\sigma;\beta/\sqrt{d},\mathbf{A}+\mathbf{J}_{ij}\right)\right]. \qquad (4.5)\end{aligned}$$

Here in the second equality we invoked Lemma 24. By Lemma 25, for some $\xi_{\sigma,i,j}\in[0,1]$ we have

$$\begin{aligned}&\left|\mathbf{E}\left[\sum_{\sigma\in\{-1,1\}^n}\sum_{i,j=1}^{n}\sigma(i)\sigma(j)\mu\left(\sigma;\beta/\sqrt{d},\mathbf{A}+\mathbf{J}_{ij}\right)\right]\right.\\ &\qquad\left.-\,\mathbf{E}\left[\sum_{\sigma\in\{-1,1\}^n}\sum_{i,j=1}^{n}\sigma(i)\sigma(j)\left(\mu\left(\sigma;\beta/\sqrt{d},\mathbf{A}\right)+\frac{\partial}{\partial\mathbf{A}_{ij}}\mu\left(\sigma;\beta/\sqrt{d},\mathbf{A}\right)\right)\right]\right|\\ &\le\frac{3\beta^2}{d}\cdot\mathbf{E}\left[\sum_{\sigma\in\{-1,1\}^n}\sum_{i,j=1}^{n}\mu\left(\sigma;\beta/\sqrt{d},\mathbf{A}+\xi_{\sigma,i,j}\mathbf{J}_{ij}\right)\right]\\ &\le\frac{3\beta^2}{d}\cdot n^2\cdot\mathrm{e}^{2\beta/\sqrt{d}}\cdot\mathbf{E}\left[\sum_{\sigma\in\{-1,1\}^n}\mu\left(\sigma;\beta/\sqrt{d},\mathbf{A}\right)\right]\\ &=\frac{3\beta^2}{d}\cdot n^2\cdot\mathrm{e}^{2\beta/\sqrt{d}}, \qquad (4.6)\end{aligned}$$

where the last inequality is due to Proposition 26. We observe that

$$\sum_{\sigma\in\{-1,1\}^n}\sum_{i,j=1}^{n}\sigma(i)\sigma(j)\mu\left(\sigma;\beta/\sqrt{d},\mathbf{A}\right) = n^2\cdot\left\langle\left(\frac{1}{n}\sum_{i=1}^{n}\sigma(i)\right)^2\right\rangle. \qquad (4.7)$$

By Proposition 23, we also have

$$\sum_{\sigma\in\{-1,1\}^n}\sum_{i,j=1}^{n}\sigma(i)\sigma(j)\frac{\partial}{\partial\mathbf{A}_{ij}}\mu\left(\sigma;\beta/\sqrt{d},\mathbf{A}\right) = -\frac{\beta}{\sqrt{d}}\cdot n^2\left(1-\langle R_{1,2}^2\rangle\right). \qquad (4.8)$$

The lemma follows by combining (4.5), (4.6), (4.7), and (4.8). ◀

The next proposition deals with the first term in the above lemma. Note that this term is trivially bounded in the bisection models for all $\beta\in\mathbb{R}$.

▶ **Proposition 28.** *For every $\beta \geq 0$, we have*

$$\lim_{d\to\infty} \sqrt{d} \cdot \limsup_{n\to\infty} \mathbf{E}\left[\left\langle \left(\frac{1}{n}\sum_{i=1}^{n}\sigma(i)\right)^2 \right\rangle_{\beta/\sqrt{d},\mathbf{A}}\right] = 0.$$

Proof. Fix an arbitrary $\epsilon > 0$ and let $S(n,\epsilon) = \{\sigma \in \{-1,1\}^n \mid -\epsilon n \leq \sum_{i=1}^n \sigma(i) \leq \epsilon n\}$. We have

$$\begin{aligned}
&\mathbf{E}\left[\left\langle \left(\frac{1}{n}\sum_{i=1}^{n}\sigma(i)\right)^2 \right\rangle_{\beta/\sqrt{d},\mathbf{A}}\right] \\
&= \mathbf{E}\left[\sum_{\sigma\in\{-1,1\}^n}\left(\frac{1}{n}\sum_{i=1}^{n}\sigma(i)\right)^2 \cdot \mu\left(\sigma;\beta/\sqrt{d},\mathbf{A}\right)\right] \\
&\leq \mathbf{E}\left[\sum_{\sigma\in S(n,\epsilon/d^{1/4})}\left(\frac{\epsilon}{d^{1/4}}\right)^2 \cdot \mu\left(\sigma;\beta/\sqrt{d},\mathbf{A}\right)\right] \\
&\qquad + \mathbf{E}\left[\sum_{\sigma\in\{-1,1\}^n\setminus S(n,\epsilon/d^{1/4})} 1 \cdot \mu\left(\sigma;\beta/\sqrt{d},\mathbf{A}\right)\right] \\
&\leq \left(\frac{\epsilon}{d^{1/4}}\right)^2 + \mathbf{E}\left[\sum_{\sigma\in\{-1,1\}^n\setminus S(n,\epsilon/d^{1/4})} \mu\left(\sigma;\beta/\sqrt{d},\mathbf{A}\right)\right]. \qquad (4.9)
\end{aligned}$$

Recall that $A_n = \{\sigma \in \{-1,1\}^n : |\sum_{i=1}^n \sigma(i)| \leq 1\}$. Given $\sigma \in \{-1,1\}^n\setminus S(n,\epsilon/d^{1/4})$, take a $\tau \in A_n$ such that $R_{1,2}(\tau,\sigma) = \max_{\sigma'\in A_n} R_{1,2}(\sigma',\sigma)$. By Poisson concentration properties (see full version for details), we have that there exists a constant $C = C(\epsilon) > 0$ such that with probability $1 - 2\cdot\exp(-Cd^{1/4}n)$ we have $H_d(\sigma) - H_d(\tau) \geq \frac{\epsilon^2\sqrt{d}n}{4}$, and consequently

$$\mu(\sigma;\beta/\sqrt{d},\mathbf{A}) \leq \mu(\tau;\beta/\sqrt{d},\mathbf{A}) \cdot \exp\left(-\frac{\beta\epsilon^2 n}{4}\right).$$

By taking expectations, we obtain

$$\begin{aligned}
\mathbf{E}[\mu(\sigma;\beta/\sqrt{d},\mathbf{A})] &\leq \mathbf{E}[\mu(\tau;\beta/\sqrt{d},\mathbf{A})] \cdot \exp\left(-\frac{\beta\epsilon^2 n}{4}\right) + 2\cdot\exp(-Cd^{1/4}n) \\
&\leq \frac{1}{|A_n|}\cdot\exp\left(-\frac{\beta\epsilon^2 n}{4}\right) + 2\cdot\exp(-Cd^{1/4}n). \qquad (4.10)
\end{aligned}$$

Here in the last inequality we used the fact that $\mathbf{E}[\mu(\tau;\beta/\sqrt{d},\mathbf{A})]$ is the same for all $\tau \in A_n$.

Combining (4.9) and (4.10), we obtain

$$\sqrt{d}\cdot\mathbf{E}\left[\left\langle \left(\frac{1}{n}\sum_{i=1}^{n}\sigma(i)\right)^2 \right\rangle_{\beta/\sqrt{d},\mathbf{A}}\right] \leq \epsilon^2 + \sqrt{d}\cdot 2^n\left(\frac{1}{|A_n|}\cdot\exp\left(-\frac{\beta\epsilon^2 n}{4}\right) + 2\exp(-Cd^{1/4}n)\right).$$

If d is sufficiently large, the above gives

$$\sqrt{d}\cdot\limsup_{n\to\infty}\mathbf{E}\left[\left\langle \left(\frac{1}{n}\sum_{i=1}^{n}\sigma(i)\right)^2 \right\rangle_{\beta/\sqrt{d},\mathbf{A}}\right] \leq \epsilon^2.$$

The proposition follows since ϵ is chosen arbitrarily. ◀

We are now ready to prove Proposition 9.

Proof of Proposition 9. For part (a), by Lemma 22, Lemma 27, and Proposition 28, we have

$$\begin{aligned}
&\lim_{d\to\infty} \limsup_{n\to\infty} \frac{1}{n} \left| \mathop{\mathbf{E}}_{\mathbf{g}} \left[\langle H_{\mathrm{SK}}(\sigma) \rangle_{\beta,\mathbf{g}} \right] - \frac{1}{\sqrt{d}} \mathop{\mathbf{E}}_{\mathbf{A}} \left[\langle H_d(\sigma) \rangle_{\beta/\sqrt{d},\mathbf{A}} \right] \right| \\
&= \lim_{d\to\infty} \left(\frac{\beta}{2} \limsup_{n\to\infty} \left| \mathop{\mathbf{E}}_{\mathbf{g}} \left[1 - \langle R_{1,2}^2 \rangle_{\beta,\mathbf{g}} \right] - \mathop{\mathbf{E}}_{\mathbf{A}} \left[1 - \langle R_{1,2}^2 \rangle_{\frac{\beta}{\sqrt{d}},\mathbf{A}} \right] \right| \right. \\
&\qquad \left. + \frac{\sqrt{d}}{2} \mathop{\mathbf{E}} \left[\left\langle \left(\frac{1}{n} \sum_{i=1}^{n} \sigma(i) \right)^2 \right\rangle_{\frac{\beta}{\sqrt{d}},\mathbf{A}} \right] + O_d\left(\frac{1}{\sqrt{d}} \right) \right) \\
&= \frac{\beta}{2} \cdot \lim_{d\to\infty} \limsup_{n\to\infty} \left| \mathop{\mathbf{E}}_{\mathbf{g}} \left[\langle R_{1,2}^2 \rangle_{\beta,\mathbf{g}} \right] - \mathop{\mathbf{E}}_{\mathbf{A}} \left[\langle R_{1,2}^2 \rangle_{\frac{\beta}{\sqrt{d}},\mathbf{A}} \right] \right|.
\end{aligned}$$

Part (a) then follows by applying Proposition 8. Part (b) follows in a similar manner and we omit the details. ◀

5 Disorder chaos for sparse models

Let $\mathbf{g}, \mathbf{g}_t, \mathbf{A}, \mathbf{A}_t$ be random matrices as defined in Section 2 and let $S \subseteq [-1,1]$. Recall that we also defined the following short-hand notation:

$$\begin{aligned}
Z^{S}_{\mathbf{g},\mathbf{g}_t} &= \sum_{\substack{\sigma_1,\sigma_2 \in \{-1,1\}^n \\ R_{1,2}(\sigma_1,\sigma_2)\in S}} \exp\left(-\beta(H(\sigma_1;\mathbf{g}) + H(\sigma_2;\mathbf{g}_t))\right), \\
Z^{S,\mathsf{bis}}_{\mathbf{g},\mathbf{g}_t} &= \sum_{\substack{\sigma_1,\sigma_2 \in A_n \\ R_{1,2}(\sigma_1,\sigma_2)\in S}} \exp\left(-\beta(H(\sigma_1;\mathbf{g}) + H(\sigma_2;\mathbf{g}_t))\right), \\
Z^{S}_{\mathbf{A},\mathbf{A}_t} &= \sum_{\substack{\sigma_1,\sigma_2 \in \{-1,1\}^n \\ R_{1,2}(\sigma_1,\sigma_2)\in S}} \exp\left(-\frac{\beta}{\sqrt{d}} \cdot (H(\sigma_1;\mathbf{A}) + H(\sigma_2;\mathbf{A}_t))\right), \\
Z^{S,\mathsf{bis}}_{\mathbf{A},\mathbf{A}_t} &= \sum_{\substack{\sigma_1,\sigma_2 \in A_n \\ R_{1,2}(\sigma_1,\sigma_2)\in S}} \exp\left(-\frac{\beta}{\sqrt{d}} \cdot (H(\sigma_1;\mathbf{A}) + H(\sigma_2;\mathbf{A}_t))\right).
\end{aligned}$$

It is known that the SK model exhibits disorder chaos at any temperature.

▶ **Theorem 29** (Theorem 9 in [15]). *Let $\beta \in \mathbb{R}$, $t > 0$. Fix an arbitrary $\epsilon > 0$. Let $I_\epsilon = [-1,-\epsilon] \cup [\epsilon,1]$. There exists some constant $K > 0$ such that for every $n \in \mathbb{N}$,*

$$\mathbf{E}\left[\frac{Z^{I_\epsilon}_{\mathbf{g},\mathbf{g}_t}}{Z^{[-1,1]}_{\mathbf{g},\mathbf{g}_t}} \right] \le K \cdot \exp(-n/K).$$

Theorem 29 immediately implies that the overlap of two configurations sampled from the coupled system $\mu_{\mathbf{g},\mathbf{g}_t}$ is nearly zero.

▶ **Corollary 30.** *If $t > 0$, then*

$$\lim_{n\to\infty} \mathbf{E}\left[\langle R_{1,2}^2 \rangle_{\beta,\mathbf{g},\mathbf{g}_t} \right] = 0.$$

Proof. For any $\epsilon > 0$, we have

$$\begin{aligned}\mathbf{E}\left[\langle R_{1,2}^2\rangle_{\beta,\mathbf{g},\mathbf{g}_t}\right] &\le \epsilon^2 \cdot \mathbf{E}\left[\frac{Z_{\mathbf{g},\mathbf{g}_t}^{[-\epsilon,\epsilon]}}{Z_{\mathbf{g},\mathbf{g}_t}^{[-1,1]}}\right] + 1 \cdot \mathbf{E}\left[\frac{Z_{\mathbf{g},\mathbf{g}_t}^{I_\epsilon}}{Z_{\mathbf{g},\mathbf{g}_t}^{[-1,1]}}\right] \\ &\le \epsilon^2 + K \cdot \exp(-n/K).\end{aligned}$$

Here we used Theorem 29 and the fact that $Z_{\mathbf{g},\mathbf{g}_t}^{[-\epsilon,\epsilon]} \le Z_{\mathbf{g},\mathbf{g}_t}^{[-1,1]}$. It follows that

$$\limsup_{n\to\infty} \mathbf{E}\left[\langle R_{1,2}^2\rangle_{\beta,\mathbf{g},\mathbf{g}_t}\right] \le \epsilon^2$$

for any arbitrary $\epsilon > 0$, and the corollary follows. ◀

The exponentially small fraction in Theorem 29 can be translated into a constant gap between the free energy of coupled and uncoupled systems.

▶ **Proposition 31.** *For every $\epsilon > 0$, there exists some constant $K > 0$ such that*

$$\limsup_{n\to\infty}\left(\frac{1}{n}\mathbf{E}\left[\log Z_{\mathbf{g},\mathbf{g}_t}^{I_\epsilon}\right] - \frac{1}{n}\mathbf{E}\left[\log Z_{\mathbf{g},\mathbf{g}_t}^{[-1,1]}\right]\right) \le -\frac{1}{K}.$$

Proof. By Theorem 29 and Jensen's inequality

$$\begin{aligned}\frac{1}{n}\mathbf{E}\left[\log Z_{\mathbf{g},\mathbf{g}_t}^{I_\epsilon}\right] - \frac{1}{n}\mathbf{E}\left[\log Z_{\mathbf{g},\mathbf{g}_t}^{[-1,1]}\right] &= \frac{1}{n}\mathbf{E}\left[\log \frac{Z_{\mathbf{g},\mathbf{g}_t}^{I_\epsilon}}{Z_{\mathbf{g},\mathbf{g}_t}^{[-1,1]}}\right] \\ &\le \frac{1}{n}\log \mathbf{E}\left[\frac{Z_{\mathbf{g},\mathbf{g}_t}^{I_\epsilon}}{Z_{\mathbf{g},\mathbf{g}_t}^{[-1,1]}}\right] \le -\frac{1}{K} + O\left(\frac{1}{n}\right).\end{aligned}$$

The proposition follows by taking n to infinity. ◀

The following theorem, which is a generalized version of Lemma 7, allows us to transfer this free energy gap to sparse models.

▶ **Theorem 32** (See e.g. [17, 16])**.** *For any $\beta \in \mathbb{R}$,*

$$\left|\frac{1}{n}\mathbf{E}\left[\log Z_{\mathbf{g},\mathbf{g}_t}^{I_\epsilon,\mathsf{bis}}\right] - \frac{1}{n}\mathbf{E}\left[\log Z_{\mathbf{A},\mathbf{A}_t}^{I_\epsilon,\mathsf{bis}}\right]\right| \le O_d\left(\frac{1}{\sqrt{d}}\right) + \sqrt{d}\cdot o_n(1).$$

We are now ready to prove Lemma 14.

Proof of Lemma 14. As before, we only prove part (a). Fix some arbitrary $\epsilon > 0$. By Theorem 16, for both $S = I_\epsilon$ and $S = [-1,1]$ we have

$$\lim_{n\to\infty}\frac{1}{n}\left(\mathbf{E}\left[\log Z_{\mathbf{g},\mathbf{g}_t}^{S,\mathsf{bis}}\right] - \mathbf{E}\left[\log Z_{\mathbf{g},\mathbf{g}_t}^{S}\right]\right) = 0.$$

By Proposition 31, for some constant $K = K(\epsilon) > 0$, we have

$$\limsup_{n\to\infty}\frac{1}{n}\left(\mathbf{E}\left[\log Z_{\mathbf{g},\mathbf{g}_t}^{I_\epsilon,\mathsf{bis}}\right] - \mathbf{E}\left[\log Z_{\mathbf{g},\mathbf{g}_t}^{[-1,1],\mathsf{bis}}\right]\right) \le -\frac{1}{K}.$$

We can then use Theorem 32 to transfer this gap to the sparse model and obtain

$$\lim_{d\to\infty}\limsup_{n\to\infty}\frac{1}{n}\left(\mathbf{E}\left[\log Z_{\mathbf{A},\mathbf{A}_t}^{I_\epsilon,\mathsf{bis}}\right] - \mathbf{E}\left[\log Z_{\mathbf{A},\mathbf{A}_t}^{[-1,1],\mathsf{bis}}\right]\right) \le -\frac{1}{K}.$$

By Theorem 19, we then have

$$\lim_{d\to\infty} \limsup_{n\to\infty} \frac{1}{n}\left(\mathbf{E}\left[\log Z^{I_\epsilon}_{\mathbf{A},\mathbf{A}_t}\right] - \mathbf{E}\left[\log Z^{[-1,1]}_{\mathbf{A},\mathbf{A}_t}\right]\right) \leq -\frac{1}{K}.$$

This means that if d is sufficiently large, then for all sufficiently large n we have

$$\frac{1}{n}\mathbf{E}\left[\log Z^{I_\epsilon}_{\mathbf{A},\mathbf{A}_t}\right] - \frac{1}{n}\mathbf{E}\left[\log Z^{[-1,1]}_{\mathbf{A},\mathbf{A}_t}\right] \leq -\frac{1}{2K}.$$

By Lemma 18, we have that with probability at least $1 - o_n(1)$

$$\frac{1}{n}\log Z^{I_\epsilon}_{\mathbf{A},\mathbf{A}_t} - \frac{1}{n}\log Z^{[-1,1]}_{\mathbf{A},\mathbf{A}_t} \leq -\frac{1}{4K},$$

which rearranges to

$$\frac{Z^{I_\epsilon}_{\mathbf{A},\mathbf{A}_t}}{Z^{[-1,1]}_{\mathbf{A},\mathbf{A}_t}} \leq \exp\left(-\frac{n}{4K}\right).$$

It follows that

$$\mathbf{E}[\langle R^2_{1,2}\rangle_{\beta/\sqrt{d},\mathbf{A},\mathbf{A}_t}] \leq \epsilon^2 + \mathbf{E}\left[\frac{Z^{I_\epsilon}_{\mathbf{A},\mathbf{A}_t}}{Z^{[-1,1]}_{\mathbf{A},\mathbf{A}_t}}\right] \leq \epsilon^2 + \exp\left(-\frac{n}{4K}\right) + o_n(1).$$

By taking n to infinity, we have $\limsup_{n\to\infty} \mathbf{E}[\langle R^2_{1,2}\rangle_{\beta/\sqrt{d},\mathbf{A},\mathbf{A}_t}] \leq \epsilon^2$. This completes the proof. ◀

References

1. Ahmed El Alaoui and David Gamarnik. Hardness of sampling solutions from the Symmetric Binary Perceptron, July 2024. arXiv:2407.16627 [cs, math]. doi:10.48550/arXiv.2407.16627.
2. Ahmed El Alaoui, Andrea Montanari, and Mark Sellke. Local algorithms for Maximum Cut and Minimum Bisection on locally treelike regular graphs of large degree, November 2021. arXiv:2111.06813 [math-ph]. doi:10.48550/arXiv.2111.06813.
3. Ahmed El Alaoui, Andrea Montanari, and Mark Sellke. Optimization of mean-field spin glasses. *The Annals of Probability*, 49(6):2922–2960, November 2021. Publisher: Institute of Mathematical Statistics. doi:10.1214/21-AOP1519.
4. Ahmed El Alaoui, Andrea Montanari, and Mark Sellke. Sampling from the Sherrington-Kirkpatrick Gibbs measure via algorithmic stochastic localization, March 2022. arXiv:2203.05093 [cond-mat]. doi:10.48550/arXiv.2203.05093.
5. Nima Anari, Vishesh Jain, Frederic Koehler, Huy Tuan Pham, and Thuy-Duong Vuong. Entropic independence I: Modified log-Sobolev inequalities for fractionally log-concave distributions and high-temperature Ising models. *arXiv preprint arXiv:2106.04105*, 2021.
6. Nima Anari, Frederic Koehler, and Thuy-Duong Vuong. Trickle-down in localization schemes and applications. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, pages 1094–1105, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3618260.3649622.
7. Roland Bauerschmidt and Thierry Bodineau. A very simple proof of the LSI for high temperature spin systems. *Journal of Functional Analysis*, 276(8):2582–2588, 2019.
8. Roland Bauerschmidt, Thierry Bodineau, and Benoit Dagallier. Kawasaki dynamics beyond the uniqueness threshold. *arXiv preprint arXiv:2310.04609*, 2023. doi:10.48550/arXiv.2310.04609.
9. Ivona Bezáková, Daniel Štefankovič, Vijay V Vazirani, and Eric Vigoda. Accelerating simulated annealing for the permanent and combinatorial counting problems. *SIAM Journal on Computing*, 37(5):1429–1454, 2008. doi:10.1137/050644033.

10 Charlie Carlson, Ewan Davies, Alexandra Kolla, and Will Perkins. Computational thresholds for the fixed-magnetization Ising model. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1459–1472, 2022. doi:10.1145/3519935.3520003.

11 Michael Celentano. Sudakov–Fernique post-AMP, and a new proof of the local convexity of the TAP free energy. *The Annals of Probability*, 52(3):923–954, 2024. doi:10.1214/23-AOP1675.

12 Sourav Chatterjee. Disorder chaos and multiple valleys in spin glasses. *arXiv preprint*, 2009. arXiv:0907.3381.

13 Antares Chen, Neng Huang, and Kunal Marwaha. Local algorithms and the failure of log-depth quantum advantage on sparse random CSPs, October 2023. arXiv:2310.01563 [quant-ph]. doi:10.48550/arXiv.2310.01563.

14 Louis H. Y. Chen. Poisson Approximation for Dependent Trials. *The Annals of Probability*, 3(3):534–545, June 1975. Publisher: Institute of Mathematical Statistics. doi:10.1214/aop/1176996359.

15 Wei-Kuo Chen. Variational representations for the Parisi functional and the two-dimensional Guerra–Talagrand bound. *The Annals of Probability*, 45(6A):3929–3966, November 2017. Publisher: Institute of Mathematical Statistics. doi:10.1214/16-AOP1154.

16 Wei-Kuo Chen, David Gamarnik, Dmitry Panchenko, and Mustazee Rahman. Suboptimality of local algorithms for a class of max-cut problems. *The Annals of Probability*, 47(3):1587–1618, May 2019. Publisher: Institute of Mathematical Statistics. doi:10.1214/18-AOP1291.

17 Wei-Kuo Chen and Dmitry Panchenko. Disorder chaos in some diluted spin glass models. *The Annals of Applied Probability*, 28(3):1356–1378, June 2018. doi:10.1214/17-AAP1331.

18 Yuansi Chen and Ronen Eldan. Localization schemes: A framework for proving mixing bounds for Markov chains. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 110–122. IEEE, 2022.

19 Amin Coja-Oghlan, Philipp Loick, Balázs F. Mezei, and Gregory B. Sorkin. The Ising Antiferromagnet and Max Cut on Random Regular Graphs. *SIAM Journal on Discrete Mathematics*, 36(2):1306–1342, June 2022. Publisher: Society for Industrial and Applied Mathematics. doi:10.1137/20M137999X.

20 Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 84(6):066106, 2011.

21 Amir Dembo, Andrea Montanari, and Subhabrata Sen. Extremal cuts of sparse random graphs. *The Annals of Probability*, 45(2):1190–1217, March 2017. Publisher: Institute of Mathematical Statistics. doi:10.1214/15-AOP1084.

22 Ronen Eldan, Frederic Koehler, and Ofer Zeitouni. A spectral condition for spectral gap: fast mixing in high-temperature Ising models. *Probability theory and related fields*, 182(3):1035–1051, 2022.

23 Francesco Guerra. Broken replica symmetry bounds in the mean field spin glass model. *Communications in mathematical physics*, 233:1–12, 2003.

24 Chris Jones, Kunal Marwaha, Juspreet Singh Sandhu, and Jonathan Shi. Random Max-CSPs Inherit Algorithmic Hardness from Spin Glasses. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 77:1–77:26, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2023.77.

25 Florent Krzakała, Andrea Montanari, Federico Ricci-Tersenghi, Guilhem Semerjian, and Lenka Zdeborová. Gibbs states and the set of solutions of random constraint satisfaction problems. *Proceedings of the National Academy of Sciences*, 104(25):10318–10323, 2007. doi:10.1073/PNAS.0703685104.

26 Aiya Kuchukova, Marcus Pappik, Will Perkins, and Corrine Yap. Fast and slow mixing of the Kawasaki dynamics on bounded-degree graphs. *arXiv preprint*, 2024. doi:10.48550/arXiv.2405.06209.

27 M Mézard, G Parisi, and R Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, pages 815–814, 2002.

28 Marc Mézard and Giorgio Parisi. Mean-field theory of randomly frustrated systems with finite connectivity. *Europhysics Letters*, 3(10):1067, 1987.

29 Marc Mézard and Giorgio Parisi. The Bethe lattice spin glass revisited. *The European Physical Journal B-Condensed Matter and Complex Systems*, 20:217–233, 2001.

30 Andrea Montanari. Optimization of the Sherrington–Kirkpatrick hamiltonian. *SIAM Journal on Computing*, 0(0):FOCS19–1–FOCS19–38, 2021. doi:10.1137/20M132016X.

31 Elchanan Mossel, Joe Neeman, and Allan Sly. Reconstruction and estimation in the planted partition model. *Probability Theory and Related Fields*, 162(3):431–461, August 2015. doi:10.1007/s00440-014-0576-6.

32 Dmitry Panchenko. *The Sherrington-Kirkpatrick Model*. Springer Monographs in Mathematics. Springer New York, NY, 2013.

33 Giorgio Parisi. Infinite number of order parameters for spin-glasses. *Physical Review Letters*, 43(23):1754, 1979.

34 Giorgio Parisi. A sequence of approximated solutions to the SK model for spin glasses. *Journal of Physics A: Mathematical and General*, 13(4):L115, 1980.

35 Giorgio Parisi. Order parameter for spin-glasses. *Physical Review Letters*, 50(24):1946, 1983.

36 David Sherrington and Scott Kirkpatrick. Solvable model of a spin-glass. *Physical review letters*, 35(26):1792, 1975.

37 H. Sompolinsky and Annette Zippelius. Dynamic theory of the spin-glass phase. *Phys. Rev. Lett.*, 47:359–362, August 1981. doi:10.1103/PhysRevLett.47.359.

38 Eliran Subag. Following the ground states of full-RSB spherical spin glasses. *Communications on Pure and Applied Mathematics*, 74(5):1021–1044, 2021.

39 Michel Talagrand. The Parisi Formula. *Annals of Mathematics*, 163(1):221–263, 2006. Publisher: Annals of Mathematics. URL: https://www.jstor.org/stable/20159953.

40 Michel Talagrand. Parisi measures. *Journal of Functional Analysis*, 231(2):269–286, February 2006. doi:10.1016/j.jfa.2005.03.001.

41 Michel Talagrand. *Mean Field Models for Spin Glasses: Volume I: Basic Examples*. Springer, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-15202-3.

A High Dimensional Cramer's Rule Connecting Homogeneous Multilinear Equations to Hyperdeterminants

Antoine Joux
CISPA – Helmholtz Center for Information Security, Saarbrücken, Germany

Anand Kumar Narayanan
SandboxAQ, Palo Alto, CA, USA

Abstract

We present a new algorithm for solving homogeneous multilinear equations, which are high dimensional generalisations of solving homogeneous linear equations. First, we present a linear time reduction from solving generic homogeneous multilinear equations to computing hyperdeterminants, via a high dimensional Cramer's rule. Hyperdeterminants are generalisations of determinants, associated with tensors of formats generalising square matrices. Second, we devise arithmetic circuits to compute hyperdeterminants of boundary format tensors. Boundary format tensors are those that generalise square matrices in the strictest sense. Consequently, we obtain arithmetic circuits for solving generic homogeneous boundary format multilinear equations. The complexity as a function of the input dimension varies across boundary format families, ranging from quasi-polynomial to sub exponential. Curiously, the quasi-polynomial complexity arises for families of increasing dimension, including the family of multipartite quantum systems made of d qubits and one qudit. Finally, we identify potential directions to resolve the hardness the hyperdeterminants, notably modulo prime numbers through the cryptographically significant tensor isomorphism complexity class.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases arithmetic circuits, tensors, determinants, hyperdeterminants

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.62

Funding *Antoine Joux*: was supported by the European Union's H2020 Programme (grant agreement #ERC-669891).

Acknowledgements We thank the anonymous ITCS 2025 referees for their valuable suggestions.

1 Introduction

1.1 Homogeneous multilinear systems

The familiar problem of solving homogeneous linear equations is to take a square matrix A and find a non zero vector x such that Ax is the zero vector. We devise algorithms for the natural high dimensional generalisation, which we call solving homogeneous multilinear equations. Let us rephrase what it means to solve homogeneous linear equations, to emphasise the motif that generalises. Given a square matrix A, find a pair of non zero vectors $\left(x^{(0)}, x^{(1)}\right)$ such that removing one of the vectors from the bilinear product $x^{(0)}Ax^{(1)}$ equals the zero vector. The solutions are merely pairs of non zero vectors $\left(x^{(0)}, x^{(1)}\right)$ with $x^{(0)}$ in the left kernel of A and $x^{(1)}$ in the right kernel of A. In homogeneous multilinear equations, a tensor A of dimension $r+1$ will be cast as the input in place of the square matrix. Multiplying A by the vectors $\left(x^{(0)}, x^{(1)}, \ldots, x^{(r)}\right)$ in the corresponding dimensions is a multilinear map taking this tuple of vectors to a scalar. If we remove one of the vectors from the multiplication, the result is a vector. The solution we demand is a tuple of vectors $\left(x^{(0)}, x^{(1)}, \ldots, x^{(r)}\right)$ such that removing one of the vectors from the multilinear product gives the zero vector, irrespective of


© Antoine Joux and Anand Kumar Narayanan;
licensed under Creative Commons License CC-BY 4.0



which one was removed. See definition 1 for the formal statement. We address the problem over the field of complex numbers and describe algorithms in the formalism of arithmetic circuits. Key to our approach is the hyperdeterminant.

1.2 Hyperdeterminants

The determinant of a square matrix is a homogeneous integer polynomial in the matrix entries that vanishes precisely when the matrix is singular. The hyperdeterminant is a high dimensional analogue of the determinant conceived by Cayley [3]. The hyperdeterminant is a homogeneous integer polynomial in the coordinates of the tensor that vanishes precisely when the tensor is singular. This notion of singularity is geometric and defined through projective duality. Except for a foray by Schläfli, the subject remained dormant for nearly a century and a half [22]. It was revived in the comprehensive work of Gelfand, Kapranov and Zelevinsky [11, 12], which contains most of the mathematical ingredients required in this paper.

1.2.1 Tensor formats

Just as the determinant is only defined for square matrices, the hyperdeterminant is only defined for certain formats of tensors. An $(r+1)$-dimensional tensor product of complex vector spaces of dimensions $k_0+1, k_1+1, \ldots, k_r+1$ constitutes a $(k_0+1) \times (k_1+1) \times \ldots \times (k_r+1)$ format. Say $k_0 \geq k_1 \geq \ldots \geq k_r$. The hyperdeterminant is defined for formats where the largest vector space dimension k_0 satisfies the convexity constraint $k_0 \leq k_1 + k_2 + \ldots + k_r$. Such formats generalise square matrices. Boundary formats are those satisfying the convexity constraint with equality, that is, $k_0 = k_1 + k_2 + \ldots + k_r$. The special case $r = 1$ corresponds to square matrices. Boundary formats generalise square matrices to higher dimensions in the strictest sense. To quote Gelfand, Kapranov and Zelevinsky [11], *"It is instructive to think of matrices of boundary format as proper high dimensional analogs of ordinary square matrices"*. Formats satisfying $k_0 < k_1 + k_2 + \ldots + k_r$ are called interior formats.

1.2.2 Hardness of Hyperdeterminants

We use hyperdeterminants as a means to solve homogeneous multilinear equations, but they are of intrinsic interest in complexity theory. The computational complexity of hyperdeterminants remains a mystery, either restricted to three dimensions or in general. Unlike the determinant (the two dimensional case), computing the hyperdeterminant (in three or more dimensions) is believed to be VNP-hard, yet a proof remains elusive. Testing if a given tensor is singular (has hyperdeterminant zero) is conjectured to be NP-hard [10]. Likewise, computing the hyperdeterminant is conjectured to be #P-hard in the counting model and VNP-hard in the arithmetic circuit model [10]. Several closely related three dimensional problems (such as zero testing singular values, defined for general formats in [16]) are proven to be NP, VNP or #P hard, but these instance are of formats where the hyperdeterminant is not defined. In particular, known hardness reductions to tensor problems seem to fall apart in formats satisfying the convexity constraint. Computing the combinatorial hyperdeterminant is known to be hard [10]. But the combinatorial hyperdeterminant more closely resembles the permanent and does not have the algebraic/geometric structure that underlies the hyperdeterminant. Another aspect to keep in mind is that the hyperdeterminant can have degree exponential in the size of the input, even in three dimensions. For instance, to write down a $(2n+1) \times (n+1) \times (n+1)$ boundary format tensor takes only cubic in n entries. However the degree of the hyperdeterminant is $(2n+1)!/n!^2 \approx 2^n$.

1.2.3 Hyperdeterminants and quantum information

Hyperdeterminants arise in quantum information when the amplitudes of quantum states are considered as normalised tensors in a projective space. The absolute value of the hyperdeterminant of three qubits ($r = 2, k_0 = k_1 = k_2 = 1$) is known as 3-tangle, an important entanglement measure generalising concurrence (the usual determinant) of bipartite systems [6]. In particular, the hyperdeterminant is invariant under local operations and classical communication (LOCC). Further significance of hyperdeterminants to quantum information was identified by Miyake and Wadati [18], through projective duality between separability and singularity.

1.3 Our Contribution

1.3.1 Reducing homogeneous multilinear systems to hyperdeterminants

The geometric notion of singularity of a tensor (the hyperdeterminant vanishing) is equivalent to the algebraic notion of degeneracy that ensures the existence of a solution to homogeneous multilinear equations. Therefore, we may test if there is a solution to the homogeneous multilinear equation by checking if the hyperdeterminant of the tensor is zero. This correspondence begs the question as to if the solutions of the multilinear equation can be inferred through computation of the hyperdeterminant. It is important to consider the model of computation for the hyperdeterminant. The minimal computational assumption is a black-box that computes the hyperdeterminant of a given tensor. But, it is not obvious how useful black-box access is. The exponential degree of the hyperdeterminant and the lack of obvious structure (such as sparsity) make interpolating the hyperdeterminant as a polynomial using black-box evaluations difficult. We instead consider white-box computation: an arithmetic circuit that takes tensor entries as inputs and outputs the hyperdeterminant.

In § 3, we present a reduction. Given an arithmetic circuit that computes the hyperdeterminant (for a tensor format), we build an arithmetic circuit of asymptotically the same size that solves generic multilinear equations (of the same format). The key to the reduction is a theorem of Gelfand, Kapranov and Zelevinsky relating the Segre embedding of solutions of multilinear equations with partial derivatives of the hyperdeterminant. It may be thought of as a high dimensional generalisation of Cramer's rule. We invoke the Baur-Strassen algorithm to construct arithmetic circuits for all these partial derivatives at once from the arithmetic circuit computing the hyperdeterminant, thereby completing the reduction with only linear complexity.

The qualifier "generic" in generic homogeneous multilinear equations refers to there being at most one projective solution. Geometrically, this translates to the input tensor either being non-singular or a simple singularity. That is, the tensor cannot be a zero of the hyperdeterminant of multiplicity greater than one. Weyman and Zelevinsky proved that non-generic tensors (roots of the hyperdeterminant of multiplicity greater than one) form a co-dimension one projective subvariety of singular tensors [30]. Therefore non-generic tensors fall into a Zariski closed subspace, justifying the "generic" label. It remains an open problem if the non generic case can be handled by methods similar to our reduction.

1.3.2 Computing Hyperdeterminants of boundary formats

In § 4, we devise arithmetic circuits to compute hyperdeterminants of boundary format tensors. There is one circuit for each boundary format. The tensor entries are the inputs to the arithmetic circuit. The main ingredient in the construction is a correspondence between the

hyperdeterminant of boundary format tensors and the determinant of a linear transformation connecting sections of vector bundles built from the tensor [11, Theorem 4.3] (see also, [7]). Concretely, the linear transformation is between two spaces of multihomogeneous polynomials in the coordinate ring with prescribed degrees. The square matrix of this linear transformation is of dimension equal to the degree $(k_0+1)!/(k_1!k_2!\dots k_r!)$ of the hyperdeterminant, which could range from 2^{k_0} to $(k_0+1)!$. By choosing a monomial bases for the polynomial spaces, we ensure that the matrix entries are either zero or entries from the tensor. An arithmetic circuit for computing the determinant for this square matrix yields an arithmetic circuit for computing the hyperdeterminant. The circuit complexity is the degree of the hyperdeterminant $(k_0+1)!/(k_1!k_2!\dots k_r!)$ raised to small exponent to account for determinant computation. In many cases this exponent can be taken to be the matrix multiplication exponent ω. At worst, the exponent is 4, for division-free circuits that accommodate computation over arbitrary rings, including fields of positive characteristic dividing the degree of the hyperdeterminant.

1.3.3 Complexity

The reduction and the algorithm for computing the hyperdeterminant in concert yield $O\left(\left(\frac{(k_0+1)!}{k_1!k_2!\dots k_r!}\right)^{\omega}\right)$ sized arithmetic circuits to solve generic homogeneous boundary format multilinear equations. To make sense of this complexity, consider the following two families of boundary format tensors. For the three dimensional family $(2n+1)\times(n+1)\times(n+1)$, the complexity is $O\left(((2n+1)!/n!^2)^{\omega}\right)$. This is roughly $O\left(2^{n\omega}\right)$, simply exponential in the dimension $4n+3$ of the output. This is sub-exponential in the dimension $(2n+1)(n+1)^2$ of the input. For the $d+1$ dimensional family $(d+1)\times\underbrace{2\times 2\times\ldots\times 2}_{d}$, the circuit complexity $O\left((d+1)!\right)$ is quasi polynomial in the input dimension $(d+1)2^d$. It is remarkable that a natural tensor problem without structure has a quasi polynomial time algorithm time for a family of increasing dimension. Further, this family captures $(d+1)$-partite quantum system consisting of d qubits and a qudit. The hyperdeterminant vanishing is related to the existence of a partition of the $(d+1)$-partite system across which the quantum state splits into a product [18]. For the most common d-partite setting consisting of d qubits, Schläfli's method [22] suggests an algorithm to compute hyperdeterminants that is recursive across the dimensions. A rigorous analysis of such a recursive algorithm based on Schläfli's method is left for future work.

In terms of algorithms to compare with, Gröbner basis methods can be deployed to solve homogeneous multilinear equations. But applying them naively only guarantees a solution in double exponential time (over Global fields such as $\mathbb{C}$). The performance of Gröbner basis techniques tailored to this problem warrants further investigation. For instance, determinantal structure was exploited in Gröbner basis algorithms addressing similar problems by Spaenlehauer [26, 27] and M. Safey El Din, and Ê. Schost [21]. Our results hint that there are monomial orderings for which Gröbner methods tailored to solving homogeneous multilinear equations are fast.

1.3.4 Towards proving hardness of hyperdeterminants

The mystery surrounding the hardness of computing the hyperdeterminant drew us to the problems addressed in this work. A technical difficulty in proving the hardness of the hyperdeterminants using well known techniques (such as in [16]) is the following. When one tries to embed a hard computational problem into computing hyperdeterminants of three

dimensional tensors, one of the dimensions of blows up and we land in a tensor format for which the hyperdeterminant does not exist. An important consequence of our reduction is that *if solving homogeneous multilinear equations is hard for some family boundary or interior formats, then so is computing the hyperdeterminant! Therefore, to prove the hardness of computing hyperdeterminants, it now suffices to prove the hardness of solving homogeneous multilinear equations for some family of boundary or interior formats.* To this end, an intermediate step might be to reduce solving inhomogeneous multilinear equations to solving homogeneous multilinear equations, since it seems easier to encode known NP-hard problems as the former. Further, our work suggests it may be fruitful to consider boundary formats such as $(2n+1) \times (n+1) \times (n+1)$, for they may accommodate more geometric methods.

Finally, in § 5, we investigate the possibility of proving the hardness computing hyperdeterminants modulo primes. To this end, we identify the (cryptographically significant) complexity class (**TI**) of tensor isomorphism problems as candidates to reduce from. In particular, we reduce the tensor isomorphism problem with respect to the special linear group action to computing hyperdeterminants modulo primes. But the tensor isomorphism complexity class is defined with isomorphisms under the general linear group action. It is not known if the tensor isomorphism problem with the special linear group action is **TI**-hard. If the answer turns out to be yes, then our reduction implies that computing hyperdeterminants modulo primes is cryptographically hard, in the sense that the average case **TI** instances are distinguished from random tensor pairs by the hyperdeterminant. If the hyperdeterminant were easy, those cryptosystems would be broken. Over fixed field sizes, it might be possible to prove that the tensor isomorphism problem with the special linear group action is indeed **TI**-hard, following ideas similar to [24]. We thank an anonymous referee for pointing out this possibility and leave this to future work.

2 Preliminaries: Tensor singularity and hyperdeterminants

2.1 Cayley's hyperdeterminants

Let $V_0, V_1, \ldots, V_r$ be $r+1$ vector spaces over the complex numbers $\mathbb{C}$ of respective dimensions $k_0+1, k_1+1, \ldots, k_r+1$. Fix a coordinate system $x^{(j)} = (x_0^{(j)}, x_1^{(j)}, \ldots, x_{k_j}^{(j)})$ for the j^{th}-vector space V_j, or equivalently an ordered basis for the dual V_j^*. Identify an $(r+1)$ dimensional tensor $A \in V_0^* \otimes V_1^* \otimes \ldots \otimes V_r^*$ with an $r+1$-dimensional matrix

$$A = (a_{i_0,i_1,\ldots,i_r}, 0 \leq i_0 \leq k_0, 0 \leq i_1 \leq k_1, \ldots, 0 \leq i_r \leq k_r)$$

of format $(k_0+1) \times (k_1+1) \times \ldots \times (k_r+1)$.

Square matrices are a special case ($r = 1$ and $k_0 = k_1$) and come with the familiar determinant whose vanishing characterises singularity/degeneracy. The hyperdeterminant is a multidimensional generalisation of the determinant that characterises singularity/degeneracy for tensors formats that generalise square matrices. We start with a geometric definition, equivalent analytic (singularity) and algebraic (degeneracy) formulations follow thereafter.

2.1.1 Geometric definition

Let $\mathbb{P}(V_j) \cong \mathbb{P}^{k_j}$ be the projectivisation of V_j. We need the Cartesian product of these projective spaces, yet desire that the product itself is projective. Let X be the image of the Cartesian product (purely separable tensors) $\mathbb{P}^{k_0} \times \mathbb{P}^{k_1} \times \ldots \times \mathbb{P}^{k_r}$ under the Segre embedding

$$\mathbb{P}(V_0) \times \mathbb{P}(V_1) \times \ldots \times \mathbb{P}(V_r) \hookrightarrow \mathbb{P}\left(V_0 \otimes V_1 \otimes \ldots \otimes V_r\right) \cong \mathbb{P}^{(k_0+1)(k_1+1)\ldots(k_r+1)-1}$$

$$\left(\left(x_0^{(0)} : x_1^{(0)} : \ldots : x_{k_0}^{(0)}\right), \ldots, \left(x_0^{(r)} : x_1^{(r)} : \ \ldots : x_{k_r}^{(r)}\right)\right) \tag{2.1}$$

$$\longmapsto \left(x_0^{(0)} x_0^{(1)} \ldots x_0^{(r)} : \ldots : x_{k_0}^{(0)} x_{k_1}^{(1)} \ldots x_{k_r}^{(r)}\right). \tag{2.2}$$

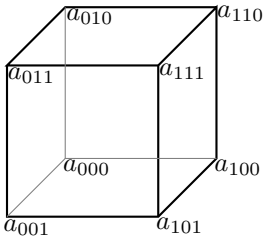
The image under the embedding is a smooth projective variety called the Segre variety, which we denote by X. Let $X^{\check{}}$ denote the projectively dual variety of X consisting of all hyperplanes in the ambient projective space $\mathbb{P}\left(V_0 \otimes V_1 \otimes \ldots \otimes V_r\right)$ tangent to X at some point. By projective duality (hyperplanes $\leftrightarrow$ points), $X^{\check{}}$ is a variety in the dual projective space $\mathbb{P}\left(V_0 \otimes V_1 \otimes \ldots \otimes V_r\right)^*$. Gelfand, Kapranov and Zelevinsky characterised precisely when $X^{\check{}}$ is a hypersurface (that is, co-dimension one). It is when the convexity condition

$$\forall\ 0 \leq j \leq r, \quad k_j \leq \sum_{\ell \neq j} k_\ell$$

holds, which we assume from here on. Being a hypersurface, the defining equation of $X^{\check{}}$ is a homogeneous polynomial in the coefficients $a_{i_1,i_2,\ldots,i_d}$, defined to be the hyperdeterminant $Det()$. It is an irreducible polynomial with integer coefficients. It can be made unique by insisting that the coefficients are co-prime and choosing a sign.

2.1.2 Example

The first example is the $r = 1$ case, of the usual 2 dimensional matrices. The convexity constraint simplifies to $k_0 = k_1$, confining to square matrices. For square matrices, the hyperdeterminant coincides with the classical determinant. The following first example in 3 dimensions goes back to Cayley [3] and the advent of hyperdeterminants. It is synonymous with the tripartite entanglement measure 3-tangle of three qubits. The hyperdeterminant of a $2 \times 2 \times 2$ format tensor A indexed by the vertices of a cube is



$$\begin{aligned} Det(A) = {} & a_{000}^2 a_{111}^2 + a_{001}^2 a_{110}^2 + a_{010}^2 a_{101}^2 + a_{011}^2 a_{100}^2 \\ & -2(a_{000}a_{001}a_{110}a_{111} + a_{000}a_{010}a_{101}a_{111} + a_{000}a_{011}a_{100}a_{111} \\ & +a_{001}a_{010}a_{101}a_{110} + a_{001}a_{011}a_{110}a_{100} + a_{010}a_{011}a_{101}a_{100}) \\ & +4(a_{000}a_{011}a_{101}a_{110} + a_{001}a_{010}a_{100}a_{111}). \end{aligned}$$

The first group of monomials correspond to the four opposing vertices across the main diagonals. The second group to the six pairs of opposing sides. The last group to the two tetrahedrons with edges on the diagonals of the faces.

2.1.3 Boundary and interior formats

Without loss of generality, assume from here on that $k_0 \geq k_1 \geq \ldots \geq k_r$. The convexity condition (which we remind, we always assume) simplifies to $k_0 \leq \sum_{\ell=1}^{r} k_\ell$. Boundary formats are those meeting the convexity constraint with equality, that is $k_0 = \sum_{\ell=1}^{r} k_\ell$. Interior formats are those satisfying the strict convexity constraint $k_0 < \sum_{\ell=1}^{r} k_\ell$.

3 Solving multilinear equations through hyperdeterminants

3.1 Hyperdeterminants, degeneracy of tensors and multilinear equations

▶ **Definition 1.** *(Solving homogeneous multilinear equations) Given a tensor A, decide if there is a $w \in X$ such that in every dimension j,*

$$\nabla_{A,j}(w) := \sum_{0 \le i_j \le k_j} \left(\sum_{\substack{0 \le i_0 \le k_0 \\ \dots \\ 0 \le i_r \le k_r}} a_{i_0,i_1,\dots,i_r} w_{i_0}^{(0)} w_{i_1}^{(1)} \dots w_{i_{j-1}}^{(j-1)} w_{i_{j+1}}^{(j+1)} \dots w_{i_r}^{(r)} \right) x_{i_j}^{(j)} = 0 \left(\in V_j^*\right).$$

The inner summation is over all dimensions except j. If such a solution w exists, find one. A tensor A is said to be degenerate if there is such a w.

Since the Segre variety X lives in the projective tensor space $\mathbb{P}(V_0 \otimes V_1 \otimes \dots \otimes V_r)$, it is expensive to write down the solution $w \in X$ as a tensor. Instead, we may output a tuple of vectors in the Cartesian space $\mathbb{P}^{k_0} \times \mathbb{P}^{k_1} \times \dots \times \mathbb{P}^{k_r}$ whose image under the Segre embedding is a solution $w \in X$.

▶ **Lemma 2** ([12, Chap 14, Prop. 1.1])**.** *A tensor A is degenerate if and only if $Det(A) = 0$.*

Proof. We sketch the proof from [12, Chap 14, Prop. 1.1] to give an impression of the connection between degeneracy and hyperdeterminants. To this end, consider the following analytic notion of singularity to complement the geometric definition of hyperdeterminants. To clarify the relation of hyperdeterminants to singularity of tensors, ask when the hyperplane $\{A = 0\}$ carved by A is in $X^\vee$. It is precisely when there is a point $x \in X$ at which the hyperplane $\{A = 0\}$ is tangent. This happens precisely when there is a point $x \in X$ such that the multilinear form (arising from the restriction of A on X)

$$f_A(x) := \sum_{\substack{0 \le i_0 \le k_0 \\ \dots \\ 0 \le i_r \le k_r}} a_{i_0,i_1,\dots,i_r} x_{i_0}^{(0)} x_{i_1}^{(1)} \dots x_{i_r}^{(r)} \text{ and all its partial derivatives } \frac{\partial f_A(x)}{\partial x_{i_j}^{(j)}}, \ \forall j, i_j$$

vanish. In particular, such an x is a singular point of the hyperplane $\{A = 0\}$. We may thus identify the hyperdeterminantal variety $X^\vee$ with singular tensors. By inspection, we see that the condition for singularity and degeneracy are the same. ◀

Therefore the decision making part of solving homogeneous multilinear equations is equivalent to testing if $Det(A) = 0$. Can hyperdeterminants be used to find solutions? We prove that they can, for the generic case of the problem, which we next define.

▶ **Definition 3** (Solving generic homogeneous multilinear equations)**.** *Given a tensor A with a promise that A is either non singular or a non singular point of $X^\vee$, solve the homogeneous multilinear equation with input A.*

We next justify why this promise version captures generic instances of the problem. As we saw before, by projective duality, there is a point $w \in X$ in the Segre variety solving the homogeneous multilinear equation corresponding to A if and only if A is in the Hyperdeterminantal variety $X^\vee$. Further, A could either be a singular point (that is, a zero of multiplicity greater than one) or a non singular point on $X^\vee$. If A is a non singular point in $X^\vee$, then there is a unique solution w, which the problem demands that we find. If A is a singular point in $X^\vee$, then we have to detect that this is the case. But, we do not have

to find a solution. Remarkably, for dimension at least three (that is, $r > 1$), excluding the interior format $2 \times 2 \times 2$, Weyman and Zelevinsky proved that the singular points of $X^\check{}$ form a co-dimension one projective sub variety [30]. Therefore, by dimension considerations, a generic singular tensor is indeed a non singular point on $X^\check{}$. The non-generic tensor inputs we abandon are in a Zariski closed subspace.

A high dimensional Cramer's rule

We reduce solving generic homogeneous multilinear equation to computing hyperdeterminants through the following characterisation of the unique solution by Gelfand, Kapranov and Zelevinsky [11, Proposition 1.2]. Let A be the input describing the homogeneous multilinear equation with the promise that A is a non singular point of $X^\check{}$. If $Det(A) \neq 0$, output that there is no solution. If $Det(A) = 0$, then the promise ensures that there is a unique solution w. Let B be a tensor of the same format as the input A, but with with entries $b_{i_0,i_1,\ldots,i_r}$ that are commuting indeterminates. The hyperdeterminant $Det(B)$ is then an integer polynomial in the indeterminates $b_{i_0,i_1,\ldots,i_r}$. Up to a normalisation factor, for all $i_0, i_1, \ldots, i_r$, the unique solution

$$w = \left(w_0^{(0)} w_0^{(1)} \ldots w_0^{(r)} : \ldots : w_{k_0}^{(0)} w_{k_1}^{(1)} \ldots w_{k_r}^{(r)}\right) \in X \subseteq \mathbb{P}\left(V_0 \otimes V_1 \otimes \ldots \otimes V_r\right)$$

in the Segre variety satisfies

$$w_{i_0}^{(0)} w_{i_1}^{(1)} \ldots w_{i_r}^{(r)} = \frac{\partial Det(B)}{\partial b_{i_0,i_1,\ldots,i_r}} \Big|_{B=A} . \tag{3.1}$$

It is too expensive to write out w as a point in the ambient tensor space $\mathbb{P}\left(V_0 \otimes V_1 \otimes \ldots \otimes V_r\right)$. Its pre-image

$$\left(\left(w_0^{(0)} : w_1^{(0)} : \ldots : w_{k_0}^{(0)}\right), \ldots, \left(w_0^{(r)} : w_1^{(r)} : \ldots : w_{k_r}^{(r)}\right)\right) \in \mathbb{P}(V_0) \times \mathbb{P}(V_1) \times \ldots \times \mathbb{P}(V_r)$$

in the Cartesian space is a succinct representation that we can explicitly write down. Returning a tuple of vectors as the solution is also in the spirit of the homogeneous linear equations that we are generalising. Since A is a non singular point of $X^\check{}$, there is at least one $(\widehat{i_0}, \widehat{i_1}, \ldots, \widehat{i_r})$ such that $\frac{\partial Det(B)}{\partial b_{\widehat{i_0},\widehat{i_1},\ldots,\widehat{i_r}}} \Big|_{B=A} \neq 0$. By equation 3.1, $w_{\widehat{i_0}}^{(0)}, w_{\widehat{i_1}}^{(1)}, \ldots, w_{\widehat{i_r}}^{(r)}$ are all non zero, which allows us to set them all to one as normalisation,

$$w_{\widehat{i_0}}^{(0)} = w_{\widehat{i_1}}^{(1)} = \ldots, w_{\widehat{i_r}}^{(r)} = 1.$$

With one non zero coordinate in each vector set to one, the i_j-th coordinate of the j-th vector is given by

$$w_{i_j}^{(j)} = \frac{\partial Det(B)}{\partial b_{\widehat{i_0},\ldots,\widehat{i_{j-1}},i_j,\widehat{i_{j+1}},\ldots,\widehat{i_r}}} \Big|_{B=A} . \tag{3.2}$$

Given an arithmetic circuit to compute the hyperdeterminant (for the format of A), the Baur-Strassen algorithm [1] constructs an arithmetic circuit that computes all $k_0 k_1 \ldots k_r$ of the partial derivatives

$$\left(\frac{\partial Det(B)}{\partial b_{i_0,i_1,\ldots,i_r}} \Big|_{B=A}, 0 \leq j \leq r, 1 \leq i_0 \leq k_0, 1 \leq i_1 \leq k_1, \ldots, 1 \leq i_r \leq k_r\right)$$

sought (in determining non zero coordinates and in equation 3.2) at once. Remarkably, the size of this arithmetic circuit is only a small constant times that of the circuit for computing the hyperdeterminant.

▶ Remark 4. For our reduction of solving multilinear equations to hyperdeterminants to hold for quantum algorithms, we need a quantum version of Baur-Strassen that converts a quantum circuit computing the hyperdeterminant into one computing all its partial deivatives demanded by equation 3.2. While there is work on quantum versions of Baur-Stassen/automatic differentiation, they are not known to work in generality. But there is an approximate algorithm of Jordan [13] for computing the gradient that suffices if we are only interested in solving the multilinear system up to first order errors.

4 Hyperdeterminants of boundary format

In this section, we devise an algorithm that given a boundary format tensor A, computes its hyperdeterminant. The algorithm can be realised as an arithmetic circuit. Recall that for boundary formats, $k_0 = k_1 + k_2 + \ldots + k_r$. Hyperdeterminants of boundary formats have a simple interpretation as resultants of a system of multilinear forms following the "Cayley trick". Slices of A in the first dimension form a collection of $k_0 + 1$ multilinear forms

$$f_A^{(i_0)}(x) := \sum_{\substack{0 \le i_1 \le k_1 \\ \ldots \\ 0 \le i_r \le k_r}} a_{i_0, i_1, \ldots, i_r} x_{i_1}^{(1)} x_{i_2}^{(2)} \ldots x_{i_r}^{(r)}, \quad 0 \le i_0 \le k_0. \tag{4.1}$$

The hyperdeterminant $Det(A)$ vanishes precisely when the resultant of $f_A^{(i_0)}(x),\ 0 \le i_0 \le k_0$ does. For positive integers $s_1, s_2, \ldots, s_r$, let

$$S(s_1, s_2, \ldots, s_r) \cong \mathrm{Sym}^{s_1}(V_1) \otimes \mathrm{Sym}^{s_2}(V_2) \otimes \ldots \otimes \mathrm{Sym}^{s_r}(V_r)$$

denote the space of polynomials that are for each $j \in 1, 2, \ldots, d$ homogeneous of degree s_j in the coordinates $x_{i_j}^{(j)}, 0 \le i_j \le k_j$. Set $m_1 := 0$ and for $j > 1$, set $m_j := k_1 + k_2 + \ldots + k_{j-1}$. The conception of our algorithms is primarily due to the following theorem relating boundary format hyperdeterminants to determinants of large matrices built with the above slices.

▶ **Theorem 5** (Gelfand-Kapranov-Zelevinsky, [11, Theorem 4.3]). *The hyperdeterminant $Det(A)$ of a boundary format $A \in V_0^* \otimes V_1^* \otimes \ldots \otimes V_r^*$ equals (up to sign) the determinant of the linear operator*

$$\delta_A : S(m_1, m_2, \ldots, m_r)^{k_0+1} \longrightarrow S(m_1 + 1, m_2 + 1, \ldots, m_r + 1)$$

$$(g_0, g_1, \ldots, g_{k_0}) \longmapsto \sum_{i_0=0}^{k_0} f_A^{(i_0)} g_{i_0}.$$

Proof. See [11][Theorem 4.3] or [7] for proofs. ◀

The matrix of δ is indeed square. We may count monomials and verify that the dimensions

$$\dim\left(S(m_1, m_2, \ldots, m_r)^{k_0+1}\right) = \frac{(k_0+1)!}{k_1! k_2! \ldots k_r!}$$

and

$$\dim\left(S(m_1 + 1, m_2 + 1, \ldots, m_r + 1)\right) = (k_1 + k_2 + \ldots + k_r + 1)\binom{k_1 + k_2 + \ldots + k_r}{k_1, k_2, \ldots, k_r}$$

of the two spaces of polynomials are the same. By theorem 5, this count is also the degree of the hyperdeterminant

$$\deg(Det(A)) = \frac{(k_0+1)!}{k_1! k_2! \ldots k_r!} = (k_1 + k_2 + \ldots + k_r + 1)\binom{k_1 + k_2 + \ldots + k_r}{k_1, k_2, \ldots, k_r}. \tag{4.2}$$

Theorem 5 gives a determinantal identity for each choice of permutation of the vector spaces $V_1, V_2, \ldots, V_r$, which is implicit in the statement. It is not clear if there is a choice of permutation better suited to computation than the others. We now have all the ingredients to describe the hyperdeterminant computation.

Computation of the hyperdeterminant

Let A be the input tensor. Fix lexicographic ordered monomial bases P and Q respectively for $S(m_1, m_2, \ldots, m_r)$ and $S(m_1+1, m_2+1, \ldots, m_r+1)$. With bases fixed, we will also denote the matrix of δ_A by δ_A. For $p \in P, q \in Q$, the (p,q)-th entry $\delta_A^{(p,q)}$ of δ_A is either 0 or an $a_{i'_0,i'_1,\ldots,i'_r}$ for some $i'_0, i'_1, \ldots, i'_r$. Since $\log(\deg(Det(A)))$ is polynomial in k_0, the encoding $(p,q) \longmapsto 0$ or $i'_0, i'_1, \ldots, i'_r$ is easy to compute in time polynomial in k_0. This encoding transforms an arithmetic circuit for computing the determinant of a $\frac{(k_0+1)!}{k_1!k_2!\ldots k_r!} \times \frac{(k_0+1)!}{k_1!k_2!\ldots k_r!}$ square matrix into an arithmetic circuit to compute the hyperdeterminant. The complexity of the circuit is $O\left(\left(\frac{(k_0+1)!}{k_1!k_2!\ldots k_r!}\right)^4\right)$, using a division-free circuit for the determinant [17]. We conclude with an illustrative example of the determinantal identity underlying the hyperdeterminant computation.

Example

Let $A = (a_{i_0,i_1,i_2})$ be a tensor of the simplest three dimensional boundary format, namely $3 \times 2 \times 2$. Through lexicographic monomial orderings, fix ordered bases $\left(x_0^{(2)}, x_1^{(2)}\right)$ of $S(0,1)$ and $\left(x_0^{(1)}x_0^{(2)}x_0^{(2)}, x_0^{(1)}x_0^{(2)}x_1^{(2)}, x_0^{(1)}x_1^{(2)}x_1^{(2)}, x_1^{(1)}x_0^{(2)}x_0^{(2)}, x_1^{(1)}x_0^{(2)}x_1^{(2)}, x_1^{(1)}x_1^{(2)}x_1^{(2)}\right)$ of $S(1,2)$. Then

$$\delta_A : S(0,1)^3 \longrightarrow S(1,2)$$

corresponds to the matrix

$$\begin{pmatrix} a_{000} & 0 & a_{100} & 0 & a_{200} & 0 \\ a_{001} & 0 & a_{101} & 0 & a_{201} & 0 \\ a_{010} & a_{000} & a_{110} & a_{100} & a_{210} & a_{200} \\ a_{011} & a_{001} & a_{111} & a_{101} & a_{211} & a_{201} \\ 0 & a_{010} & 0 & a_{110} & 0 & a_{210} \\ 0 & a_{011} & 0 & a_{111} & 0 & a_{211} \end{pmatrix}.$$

5 On the hardness of computing hyperdeterminants modulo primes

Let p denote a prime. Recall from § 2 that for valid tensor formats, $Det()$ is a polynomial with integer coefficients, that is primitive and uniquely defined by choosing a sign. We prove that given a tensor $A = (a_{i_0,i_1,i_2}, 0 \leq i_0, i_1, i_2 \leq n)$ of "cubical" format $(n+1)\times(n+1)\times(n+1)$ with integer coordinates (that is, $\forall (i_0,i_1,i_2), a_{i_0,i_1,i_2} \in \mathbb{Z}$), it is hard to compute $Det(A)$ mod p, under certain post-quantum cryptographic assumptions.

Let $\mathbb{F}_p$ denote the finite field with a prime number p element. Fix an algebraic closure $\bar{\mathbb{F}}_p$ of $\mathbb{F}_p$. The theory of hyperdeterminants is yet to be worked out rigorously over $\bar{\mathbb{F}}_p$, unlike the theory over complex numbers developed by Gelfand, Kapranov and Zelevinsky. The key issue is that the geometric theory of projective duality used in the very definition of hyperdeterminants does not in general translate to positive characteristic. Thankfully, in our particular context, the duality used to define and prove the existence of hyperdeterminants is

proven to hold true over $\bar{\mathbb{F}}_p$ for all primes p by Kaji [14]. We will work under the assumption that the basic properties of hyperdeterminants from [11] translate to $\bar{\mathbb{F}}_p$. In particular, the hyperdeterminant over $\bar{\mathbb{F}}_p$ is $Det(A)$ mod p, the hyperdeterminant over the integers modulo p. We defer rigorous proofs to later works, focusing instead on sketching the main ideas in the reduction.

We next state the computational problem of computing hyperdeterminants modulo primes. To control the length of the input, without loss of generality, we may assume that the coordinates $a_{i_0,i_1,i_2} \in \mathbb{F}_p$ instead of being integers. It is therefore convenient to think of the input as being

$$A = (a_{i_0,i_1,i_2}, 0 \leq i_0, i_1, i_2 \leq n) \in (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^*$$

of "cubical" format $(n+1) \times (n+1) \times (n+1)$ over $\mathbb{F}_p$. Let $Det_{n+1,p}()$ denote the polynomial over $\mathbb{F}_p$ obtained by reducing $Det()$ modulo p. In particular,

$$Det_{n+1,p}() \in \mathbb{F}_p[a_{i_0,i_1,i_2}, 0 \leq i_0, i_1, i_2 \leq n]$$

is a non zero polynomial in the coordinate ring of A. As a convention, we will use $Det_{n+1,p}()$ to denote the polynomial and $Det_{n+1,p}(A)$ to denote its evaluation at a tensor A. The computational problem in question is then given a tensor A to compute $Det_{n+1,p}(A) \in \mathbb{F}_p$. Associated with a cubical tensor $A \in (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^*$ is the trilinear form

$$f_A(x^{(0)}, x^{(1)}, x^{(2)}) := \sum_{0 \leq i_0, i_1, i_2 \leq n} a_{i_0,i_1,i_2} x^{(0)}_{i_0} x^{(1)}_{i_1} x^{(2)}_{i_2}.$$

Triples $(G_0, G_1, G_2) \in (GL_{n+1}(\mathbb{F}_p))^3$ of invertible $(n+1) \times (n+1)$ matrices act on A (or equivalently f_A) in the natural way,

$$(G_0, G_1, G_2) \circ f_A := f_A(G_0 x^{(0)}, G_1 x^{(1)}, G_2 x^{(2)}).$$

Call two tensors A and B as isomorphic if there exists a triple $(G_0, G_1, G_2) \in (GL_{n+1}(\mathbb{F}_p))^3$ such that $f_B = (G_0, G_1, G_2) \circ f_A$. This is indeed an equivalence relation. In particular, it is symmetric since $f_B = (G_0, G_1, G_2) \circ f_A$ if and only if $f_A = (G_0^{-1}, G_1^{-1}, G_2^{-1}) \circ f_B$.

The Tensor Isomorphism Problem. To decide if two cubical tensors over finite fields are isomorphic is the tensor isomorphism problem, which is believed to be hard. Grochow and Qiao built an intricate web of hard problems that reduce to tensor isomorphism, including some longstanding hard problems that lay at the foundation of multivariate cryptography. Complexity theoretically, the tensor isomorphism problem is $NP \cap co - AM$, and believed to be hard on average in theory and practice [9]. The best known run time of $p^{O(n^{11/6})}$ is through Sun's p-group isomorphism algorithm [28] (in conjunction with a reduction in [8]). The promise search version, given two isomorphic A, B to find a $(G_0, G_1, G_2) \in (GL_{n+1}(\mathbb{F}_p))^3$ such that $f_B = (G_0, G_1, G_2) \circ f_A$ is also believed hard. Spurred on by this hardness, several post-quantum digital signature schemes including MEDS [5] and ALTEQ [2, 29] have recently been proposed and submitted to NIST call for post-quantum signatures, all reliant on tensor isomorphism hardness assumptions, or hardness assumptions that reduce to tensor isomorphism.

We focus our attention on MEDS which involves the exact tensor isomorphism stated in our context. The signature scheme is built as follows. First a zero-knowledge interactive protocol is constructed based on the Goldreich-Micali-Wigderson protocol for graph isomorphisms. Except, the graph isomorphism problem is recast with tensor isomorphism

problem. Then the interaction in the zero-knowledge protocol is removed using the Fiat-Shamir transformation to yield a signature scheme. We identify the following hardness assumption underlying the soundness and zero-knowledge proofs of the interactive protocol as one that easily relates to hyperdeterminants.

▶ **Assumption 6.** *Draw a tensor* $A \in (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^*$ *uniformly at random. Draw a triple* $(G_0, G_1, G_2) \in (GL_{n+1}(\mathbb{F}_p))^3$ *of invertible matrices uniformly at random and set* $f_B := (G_0, G_1, G_2) \circ f_A$. *The pair* (A, B) *is computationally indistinguishable from a uniformly random pair of tensors in* $\left((\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^*\right)^2$.

An algorithm that distinguishes such an isomorphic pair (A, B) from a uniformly random pair of tensors can be used to either recover the key in a No-Message-Attack or forge the signature in a Chosen-Message-Attack.

Unfortunately, we don't quite know how to use the hyperdeterminant to build a distinguisher for assumption 6. The reason being that the hyperdeterminant is only a relative invariant of the $(GL_{n+1}(\mathbb{F}_p))^3$ action. That is, for all $A \in (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^*$ and for all $(G_0, G_1, G_2) \in (GL_{n+1}(\mathbb{F}_p))^3$,

$$Det_{n+1,p}((G_0, G_1, G_2) \circ f_A) = C(G_0, G_1, G_2) Det_{n+1,p}(f_A),$$

where the "constant" $C(G_0, G_1, G_2)$ depends on the acting matrices. However, if we restrict to $(SL_{n+1}(\mathbb{F}_p))^3$ action, then the hyperdeterminant is an invariant, that is, for all $A \in (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^*$ and for all $(G_0, G_1, G_2) \in (SL_{n+1}(\mathbb{F}_p))^3$,

$$Det_{n+1,p}((G_0, G_1, G_2) \circ f_A) = Det_{n+1,p}(f_A).$$

We observe that there is a natural hardness reduction from the tensor isomorphism problem restricted to the $(SL_{n+1}(\mathbb{F}_p))^3$ action to the hyperdeterminant. To this end, we next write out the hardness assumption restricted to the $(SL_{n+1}(\mathbb{F}_p))^3$ action, followed by the distinguisher with blackbox access to the hyperdeterminant that reduces the tensor isomorphism problem with the $(SL_{n+1}(\mathbb{F}_p))^3$ action to the hyperdeterminant.

▶ **Assumption 7.** *Draw a tensor* $A \in (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^*$ *uniformly at random. Draw a triple* $(G_0, G_1, G_2) \in (SL_{n+1}(\mathbb{F}_p))^3$ *of determinant one matrices uniformly at random and set* $f_B := (G_0, G_1, G_2) \circ f_A$. *The pair* (A, B) *is computationally indistinguishable from a uniformly random pair of tensors in* $\left((\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^*\right)^2$.

Distinguisher. Given a pair of tensors $(A, B) \in \left((\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^*\right)^2$, decide that the input is a pair of isomorphic tensors if and only if $Det_{n+1,p}(A) = Det_{n+1,p}(B)$.

By invariance, if A and B are isomorphic, then indeed $Det_{n+1,p}(A) = Det_{n+1,p}(B)$, meaning the distinguisher is always correct when presented with two isomorphic tensors. Therefore, for our distinguisher to succeed, it suffices to prove that for a uniformly random tensor A, $det(A)$ is not concentrated on a particular value in $\mathbb{F}_p$. In the following lemma, for large enough field size p (growing exponentially with n), we prove for uniformly random A that $Det(A)$ is close to uniformly distributed. For such large p, our distinguisher succeeds with probability close to $1 - 1/p$. We take the lemma as strong evidence that there is no arithmetic obstruction to equidistribution, meaning equidistribution should hold for even for small p. The requirement on p being large is merely an artifact our proof methods and our distinguisher should succeed with probability close to $1 - 1/p$ for all p.

▶ **Lemma 8.** *Fix the tensor dimension n and a positive parameter $c > 1$. For every large enough prime $p \geq (n/2)^{\frac{c-1}{c}} 2^{\frac{(c-1)n}{c}\left(1+\frac{3}{2}\log_2 n\right)}$, and for every $u \in \mathbb{F}_p$,*

$$Prob_A\left(Det_{n+1,p}(A) = u\right) \leq \frac{1}{p^{1/c}} + o\left(\frac{1}{p^{\frac{1}{c}}}\right),$$

where the probability is taken over uniform $A \in (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^*$. For even larger primes p with $p \geq 5(n/2)^{13/3} 2^{\frac{13n}{3}\left(1+\frac{3}{2}\log_2 n\right)}$, for every $u \in \mathbb{F}_p$,

$$Prob_A\left(Det_{n+1,p}(A) = u\right) \leq \frac{1}{p} + o\left(\frac{1}{p}\right),$$

a tighter bound showing that the Hyperdeterminant values are nearly uniformly distributed.

Proof. Since the hyperdeterminant is a primitive polynomial with integer coefficients, reduction by p gives a non zero polynomial $Det_{n+1,p}() \in \mathbb{F}_p[a_{i_0,i_1,i_2}, 0 \leq i_0, i_1, i_2 \leq n]$. By [12, Cor. 2.9], its degree is bounded by

$$\deg(Det_{n+1,p}()) \leq \sum_{0\leq j\leq k/2} \frac{(j+k_1)!}{j!^3(k-2j)!} 2^{k-2j} \leq \sum_{0\leq j\leq k/2} n^{3j} 2^{n-2j} \leq \frac{n}{2} 2^{n\left(1+\frac{3}{2}\log n\right)}.$$

Further, $Det_{n+1,p}() - u$ is a non-zero polynomial with the same degree bound. Bounding the number of roots of $Det_{n+1,p}() - u$ by Schwartz-Zippel lemma [23],

$$Prob_A\left(Det_{n+1,p}(A) = u\right) \leq \frac{\deg(Det_{n+1,p}() - u)}{p}.$$

This proves the first bound in the lemma, for every choice of $c > 1$. If one desires a tighter bound towards the uniformity of the hyperdeterminant values, we can resort to the more sophisticated Lang-Weil bound [15]. The fewer the number of components of the variety generated by $Det_{n+1,p}() - u$, the better the Lang-Weil bound. Over the complex numbers, the hyperdeterminant is an absolutely irreducible integer polynomial. By analogy, $Det_{n+1,p}() - u$ is irreducible in $\bar{\mathbb{F}}_p[a_{i_0,i_1,i_2}, 0 \leq i_0, i_1, i_2 \leq n]$. The variety $Det_{n+1,p}() - u$ defines has only one component and the Lang-Weil bound implies

$$Prob_A\left(Det_{n+1,p}(A) = u\right) \leq \frac{1}{p} + O_{\deg(Det_{n+1,p}()-u)}\left(\frac{1}{p}\right).$$

But the asymptotic notation $O_{\deg(Det_{n+1,p}()-u)}()$ in the original Lang-Weil paper hides terms depending on n that are too big. Therefore, we look to the effective version of the Lang-Weil bound, optimized for irreducible hypersurfaces in [25, Thm. 2], which for $p > 5\deg(Det_{n+1,p}() - u)^{13/3}$ implies

$$Prob_A\left(Det_{n+1,p}(A) = u\right) \leq \frac{1}{p} + \frac{(\deg(Det_{n+1,p}() - u) - 1)(\deg(Det_{n+1,p}() - u) - 2)}{p^{3/2}} + \frac{5}{p^2}.$$

Therefore, for $p \geq 5(n/2)^{13/3} 2^{\frac{13n}{3}\left(1+\frac{3}{2}\log_2 n\right)} \geq 5\deg(Det_{n+1,p}() - u)^{13/3}$, the second bound claimed in the lemma is true. ◀

In light of this reduction, to prove tensor isomophism hardness of computing the hyperdeteminant, it suffices to answer the following open question in the affirmative.

▶ **Question 9.** *Is the problem of deciding if for two given tensors A and B there exists a triple $(G_0, G_1, G_2) \in (SL_{n+1}(\mathbb{F}_p))^3$ of determinant one matrices such that $f_B = (G_0, G_1, G_2) \circ f_A$ tensor isomorphism hard? In other words, is there a polynomial time reduction from the tensor isomorphism problem with $(GL_{n+1}(\mathbb{F}_p))^3$ action to that with $(SL_{n+1}(\mathbb{F}_p))^3$ action?*

We suspect that the answer to the question is yes, but also advise caution. Tensor isomorphism problems with other matrix group actions such as the orthogonal, symplectic or unitary (over complex numbers) are in practice easier than the general linear group actions [4].

▶ Remark 10. For pairs (n,p) such that $p-1$ divides $deg(Det_{n+1,p}())/(n+1)$, our distinguisher for the $(SL_{n+1}(\mathbb{F}_p))^3$ actually also works as a distinguisher for the $(GL_{n+1}(\mathbb{F}_p))^3$ action. Therefore, restricted to the pairs (n,p) satisfying $p-1$ divides $deg(Det_{n+1,p}())/(n+1)$, the hyperdeteminant is at least as hard as the cryptographically hard problem underlying MEDS.

The reason is that the relative invariance of the hyperdeterminant with respect to the $GL_{n+1}(\mathbb{F}_p)^3$ action takes the following form (c.f. proof of theorem 4.4 in [19]). For all $A \in (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^* \otimes (\mathbb{F}_p^{n+1})^*$ and for all $(G_0,G_1,G_2) \in (GL_{n+1}(\mathbb{F}_p))^3$,

$$Det_{n+1,p}((G_0,G_1,G_2)\circ f_A) = (Det(G_0)Det(G_1)Det(G_2))^{deg(Det_{n+1,p}())/(n+1)}\, Det_{n+1,p}(f_A).$$

But for p such that $p-1$ divides $deg(Det_{n+1,p}())/(n+1)$,

$$(Det(G_0)Det(G_1)Det(G_2))^{deg(Det_{n+1,p}())/(n+1)} = 1,$$

since raising a non zero element in $\mathbb{F}_p$ to the $p-1$-th power is one. Therefore, for (n,p) such that $p-1$ divides $deg(Det_{n+1,p}())/(n+1)$, the hyperdeterminant is actually an invariant of the $GL_{n+1}(\mathbb{F}_p)^3$ action, not merely a relative invariant.

Let us pause to understand the pairs (n,p) for which the reduction applies. By [12, Cor. 2.9], the degree of the cubical hyperdeterminant is

$$\deg(Det_{n+1,p}()) = \sum_{0\le j\le k/2} \frac{(j+k_1)!}{j!^3(k-2j)!}2^{k-2j} \le \sum_{0\le j\le k/2} n^{3j}2^{n-2j} \le \frac{n}{2}2^{n\left(1+\frac{3}{2}\log n\right)}.$$

Further, $\deg(Det_{n+1,p}())$ is divisible by $n+1$, although it is not apparent from the expression ([19, Theorem 4.4]). For a choice of n, there are at most finitely many p satisfying the condition $p-1$ divides $\deg(Det_{n+1,p}())/(n+1)$. Some of these primes may be exponential in n, which is consistent with the large size of primes chosen in MEDS and ALTEQ. The number of pairs (n,p) satisfying the condition should be infinite assuming some widely believed conjectures on the distribution of primes in polynomial values, but is difficult to prove unconditionally.

▶ Remark 11. Recently, Ran and Samardjiska [20] identified a potential weakness in MEDS and ALTEQ signature schemes that rely on the hardness of tensor isomorphism problems. If the public keys chosen turn out to support a certain triple of points they call "triangles", then the isomorphism problem may be easier than the generic case. They further describe Grobner basis based algorithms for detecting and finding triangles. If such triangles exist, their algorithms can strip several bits of security off the schemes. Therefore, careful analysis of the weak key issue they identified is warranted. We observe that in the language of hyperdeterminants these weak keys correspond to singular tensors and the triangles are solutions to the homogeneous multilinear equations in 1 modulo p. Further, hyperdeterminants play a curious role in analysing this critical weak key security issue. First, $Det_{n+1,p}()$ being irreducible over $\bar{\mathbb{F}}_p$ means that weak keys arise with probability roughly $\frac{1}{p}$. Therefore, increasing p to be exponential in the security parameter is an immediate remedy (at the cost of efficiency, needing slightly larger signature lengths etc.). If zero testing the hyperdeterminant (that is, given a tensor A deciding if $Det_{n+1,p}(A) = 0$) turns out to be easy, then it can be used to test and weed out the weak keys. Further, any algorithm for computing the hyperdeterminant $Det_{n+1,p}()$ can be turned into one for computing the triangles through our tensor Cramer's rule algorithm in § 3.1.

References

1 Walter Baur and Volker Strassen. The complexity of partial derivatives. *Theoretical Computer Science*, 22(3):317–330, 1983. doi:10.1016/0304-3975(83)90110-X.

2 M. Bläser, D. H. Duong, A. K. Narayanan, T. Plantard, Y. Qiao, A. Sipasseuth, and G. Tang. The alteq signature scheme: Algorithm specifications and supporting documentation, 2023. URL: https://pqcalteq.github.io/ALTEQ_spec_2023.09.18.pdf.

3 Arthur Cayley. *On the Theory of Elimination*, pages 370–374. Cambridge Library Collection – Mathematics. Cambridge University Press, 2009.

4 Zhili Chen, Joshua A. Grochow, Youming Qiao, Gang Tang, and Chuanqi Zhang. On the Complexity of Isomorphism Problems for Tensors, Groups, and Polynomials III: Actions by Classical Groups. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, volume 287 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:23, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2024.31.

5 Tung Chou, Ruben Niederhagen, Edoardo Persichetti, Tovohery Hajatiana Randrianarisoa, Krijn Reijnders, Simona Samardjiska, and Monika Trimoska. Take your meds: Digital signatures from matrix code equivalence. In Nadia El Mrabet, Luca De Feo, and Sylvain Duquesne, editors, *Progress in Cryptology – AFRICACRYPT 2023*, pages 28–52, Cham, 2023. Springer Nature Switzerland. doi:10.1007/978-3-031-37679-5_2.

6 Valerie Coffman, Joydip Kundu, and William K. Wootters. Distributed entanglement. *Phys. Rev. A*, 61:052306, April 2000. doi:10.1103/PhysRevA.61.052306.

7 Carla Dionisi and Giorgio Ottaviani. The binet–cauchy theorem for the hyperdeterminant of boundary format multi-dimensional matrices. *Journal of Algebra*, 259(1):87–94, 2003. doi:10.1016/S0021-8693(02)00537-9.

8 J. Grochow and Y. Qiao. On the complexity of isomorphism problems for tensors, groups, and polynomials iv: linear-length reductions and their applications.

9 Joshua Grochow and Youming Qiao. On the complexity of isomorphism problems for tensors, groups, and polynomials i: Tensor isomorphism-completeness. *SIAM Journal on Computing*, 52(2):568–617, 2023. doi:10.1137/21M1441110.

10 Christopher J. Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *J. ACM*, 60(6), November 2013. doi:10.1145/2512329.

11 M. Kapranov I. Gelfand and A. Zelevinsky. Hyperdeterminants. *Advances in Mathematics*, 96:226–263, 1992.

12 M. Kapranov I. Gelfand and A. Zelevinsky. *Discriminants, Resultants, and Multidimensional Determinants*. Modern Birkhäuser Classics, 1994.

13 Stephen P. Jordan. Fast quantum algorithm for numerical gradient estimation. *Phys. Rev. Lett.*, 95:050501, July 2005. doi:10.1103/PhysRevLett.95.050501.

14 H. Kaji. On the duals of segre varieties. *Geometriae Dedicata*, 99:221–229, 2003.

15 Serge Lang and André Weil. Number of points of varieties in finite fields. *American Journal of Mathematics*, 76(4):819–827, 1954. URL: http://www.jstor.org/stable/2372655.

16 Lek-Heng Lim. Singular values and eigenvalues of tensors: a variational approach. In *1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, 2005.*, pages 129–132, 2005. doi:10.1109/CAMAP.2005.1574201.

17 M. Mahajan and V. Vinay. Determinant: Combinatorics, algorithms, and complexity. *Chicago Journal of Theoretical Computer Science*, 5, 1997.

18 Akimasa Miyake and Miki Wadati. Multipartite entanglement and hyperdeterminants. *Quantum Info. Comput.*, 2(7):540–555, December 2002. doi:10.26421/QIC2.S-4.

19 Giorgio Ottaviani. *Introduction to the Hyperdeterminant and to the Rank of Multidimensional Matrices*, pages 609–638. Springer New York, New York, NY, 2013. doi:10.1007/978-1-4614-5292-8_20.

20 Lars Ran and Simona Samardjiska. Rare structures in tensor graphs – bermuda triangles for cryptosystems based on the tensor isomorphism problem. Cryptology ePrint Archive, Paper 2024/1396, 2024. URL: https://eprint.iacr.org/2024/1396.

21 Mohab Safey El Din and Éric Schost. Bit complexity for multi-homogeneous polynomial system solving—application to polynomial minimization. *Journal of Symbolic Computation*, 87:176–206, 2018. doi:10.1016/j.jsc.2017.08.001.

22 L. Schläfli. *Über die Resultante eines Systemes mehrerer algebraischer Gleichungen*, pages 9–112. Springer Basel, Basel, 1953. doi:10.1007/978-3-0348-4117-7_1.

23 J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980. doi:10.1145/322217.322225.

24 N. Sendrier and D. E. Simos. How easy is code equivalence over fq? In *International Workshop on Coding and Cryptography – WCC 2013*, 2013.

25 K. Slavov. Improved lang–weil bounds for a geometrically irreducible hypersurface over a finite field. *Canadian Mathematical Bulletin*, 66(2):654–664, 2023.

26 P-J. Spaenlehauer. *Solving multi-homogeneous and determinantal systems: algorithms, complexity, applications*. PhD thesis, Universitê Pierre et Marie Curie (Univ. Paris 6), 2012.

27 Pierre-Jean Spaenlehauer. On the complexity of computing critical points with gröbner bases. *SIAM Journal on Optimization*, 24(3):1382–1401, 2014. doi:10.1137/130936294.

28 Xiaorui Sun. Faster isomorphism for p-groups of class 2 and exponent p. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pages 433–440, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3564246.3585250.

29 Gang Tang, Dung Hoang Duong, Antoine Joux, Thomas Plantard, Youming Qiao, and Willy Susilo. Practical post-quantum signature schemes from isomorphism problems of trilinear forms. Eurocrypt 2022, 2022. URL: https://eprint.iacr.org/2022/267.

30 Jerzy Weyman and Andrei Zelevinsky. Singularities of hyperdeterminants. *Annales de l'institut Fourier*, 46(3):591–644, 1996. URL: http://eudml.org/doc/75190.

Complexity Classification of Product State Problems for Local Hamiltonians

John Kallaugher
Sandia National Laboratories, Albuquerque, NM, USA

Ojas Parekh
Sandia National Laboratories, Albuquerque, NM, USA

Kevin Thompson
Sandia National Laboratories, Albuquerque, NM, USA

Yipu Wang
Sandia National Laboratories, Albuquerque, NM, USA

Justin Yirka
Sandia National Laboratories, Albuquerque, NM, USA
The University of Texas at Austin, TX, USA

Abstract

Product states, unentangled tensor products of single qubits, are a ubiquitous ansatz in quantum computation, including for state-of-the-art Hamiltonian approximation algorithms. A natural question is whether we should expect to efficiently solve product state problems on any interesting families of Hamiltonians.

We completely classify the complexity of finding minimum-energy product states for Hamiltonians defined by any fixed set of allowed 2-qubit interactions. Our results follow a line of work classifying the complexity of solving Hamiltonian problems and classical constraint satisfaction problems based on the allowed constraints. We prove that estimating the minimum energy of a product state is in P if and only if all allowed interactions are 1-local, and NP-complete otherwise. Equivalently, any family of non-trivial two-body interactions generates Hamiltonians with NP-complete product-state problems. Our hardness constructions only require coupling strengths of constant magnitude.

A crucial component of our proofs is a collection of hardness results for a new variant of the Vector Max-Cut problem, which should be of independent interest. Our definition involves sums of distances rather than squared distances and allows linear stretches.

We similarly give a proof that the original Vector Max-Cut problem is NP-complete in 3 dimensions. This implies hardness of optimizing product states for Quantum Max-Cut (the quantum Heisenberg model) is NP-complete, even when every term is guaranteed to have positive unit weight.

2012 ACM Subject Classification Theory of computation → Quantum complexity theory

Keywords and phrases quantum complexity, quantum algorithms, local hamiltonians

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.63

Related Version *Revised Version*: https://arxiv.org/abs/2401.06725v2

Funding This work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research in Quantum Computing, Fundamental Algorithmic Research for Quantum Computing. O.P. was also supported by U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Quantum Systems Accelerator. JY was partially supported by Scott Aaronson's Simons Investigator Award.

Acknowledgements Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. This written work is authored by an employee of



© John Kallaugher, Ojas Parekh, Kevin Thompson, Yipu Wang, and Justin Yirka;
licensed under Creative Commons License CC-BY 4.0



NTESS. The employee, not NTESS, owns the right, title and interest in and to the written work and is responsible for its contents. Any subjective views or opinions that might be expressed in the written work do not necessarily represent the views of the U.S. Government. The publisher acknowledges that the U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this written work or allow others to do so, for U.S. Government purposes. The DOE will provide public access to results of federally sponsored research in accordance with the DOE Public Access Plan.

1 Introduction

Product states, unentangled tensor products of single-qubit states, have served as an effective focus for better understanding quantum phenomena. Because general quantum states cannot be described efficiently, approximation algorithms must be restricted to output some subset of states, an ansatz. Mean-field approaches are common as first steps in statistical mechanics, and recent approximation algorithms for extremal energy states of local Hamiltonians have relied on proving that product states provide good approximations in particular regimes [17, 4, 5, 18, 31, 32]. In fact, because in some natural regimes the ground states are rigorously well-approximated by product states [4], optimal approximation algorithms for local Hamiltonians on arbitrary interaction graphs must be capable of finding good product states. Understanding product state optimization is essential for understanding the complexity of Hamiltonian approximation generally.

Product states are a natural intermediate between classical and quantum states, allowing for superposition but not entanglement. Unlike general quantum states, they have succinct classical descriptions: a single-qubit pure state can be specified by two complex numbers, and an n-qubit pure product state can be specified by $2n$ complex numbers. One could consider "more quantum" intermediates, in the form of reduced states of two or more qubits. However, verifying the consistency of a set of quantum marginals is a QMA-complete problem, even for 2-qubit reduced states [27, 10]. Therefore, product states are uniquely useful when optimizing directly over state vectors.

We study the following question: for a family of Hamiltonians defined by a given set of allowed interactions, what is the complexity of computing the extremal energy over product states? Additionally, how does the complexity of optimizing over product states relate to that of optimizing over general states? For example, for a QMA-hard local Hamiltonian, must finding the optimal product state in turn be NP-hard?[1]

This question follows a long line of work classifying the complexity of constraint satisfaction problems (CSPs) based on the sets of allowed constraints, clauses, or interactions between variables. In particular, the dichotomy theorem of Schaefer [35] showed that for any set of allowed Boolean constraints, the family of CSPs is either efficiently decidable or is NP-complete. In the context of quantum problems, Cubitt and Montanaro [14] introduced a similar classification of ground state energy problems for 2-local Hamiltonians, showing that for any fixed set of allowed 2-qubit interactions, the 2-LH problem is either in P or NP-, StoqMA-, or QMA-complete (the StoqMA case relies on the concurrent work of Bravyi and Hastings [6]). We briefly survey some of this line of classical and quantum work in Related Work below.

[1] The product state problem is always in NP since product states have succinct classical descriptions with which we can compute the expected energy contribution from each local Hamiltonian term in time polynomial in the size of the term.

While the complexity of finding the extremal states (e.g. the ground state) of 2-local Hamiltonians is well understood, the complexity of finding optimal product state solutions has been only sparsely studied [22]. The only NP-hardness results for such problems are based on mapping a classical problem to a diagonal 0-1 valued Hamiltonian [39].

An additional motivation for our study is the hope of developing new methods for identifying families of local Hamiltonians for which problems involving *general* ground states are *not* hard. While a complete complexity classification for the general 2-LH problem is known, more refined attempts at classification which take into account restrictions on the sign of the weights or geometry of the system are currently incomplete [33]. Developing algorithms for product states is a "mostly classical" problem that is easier to analyze, and progress involving product states may inform our expectations regarding general states.

In this work, we completely classify the complexity of finding optimal product states for families of 2-local Hamiltonians. In fact, we find the complexity of the product state problem is fully determined by the complexity of the general local Hamiltonian problem: if the general problem is NP-hard, the product state problem is NP-complete, and otherwise it is in P. To arrive at our results, we study a variant of the Vector Max-Cut which should be of independent interest especially to the optimization community. As a corollary to our classification theorem, we give the first published proof that estimating optimal product state energies in the Quantum Max-Cut model is NP-complete, and we show hardness holds even for unweighted Hamiltonians.[2]

1.1 Our Contributions

Formal definitions are given in Section 2. A k-local Hamiltonian is a sum of Hamiltonian terms each of which only acts non-trivially on at most k qubits, analogous to k-variable Boolean clauses. k-LH denotes the problem of estimating the ground state energy (the minimum eigenvalue across all states) of a k-local Hamiltonian to inverse-polynomial additive precision. Given a set of local terms $\mathcal{S}$, $\mathcal{S}$-LH is k-LH restricted to Hamiltonians such that every term belongs to $\mathcal{S}$. Finally, prodLH and $\mathcal{S}$-prodLH are the restrictions of these problems to product states, i.e. to minimize $\langle\phi|H|\phi\rangle$ where H is the Hamiltonian and $|\phi\rangle$ ranges over tensor products of single-qubit states.[3]

2-local $\mathcal{S}$-prodLH

The classification of the general ground state energy problem by Cubitt and Montanaro [14] completely classifies $\mathcal{S}$-LH for any fixed set $\mathcal{S}$ of 2-qubit terms, showing it is either in P or it is one of NP-, StoqMA-, or QMA-complete.

In the same vein, we give a complete classification of product state complexity for families of 2-local Hamiltonians as a function of the set of allowed 2-qubit interactions. For any given set $\mathcal{S}$ of 2-qubit terms, we prove the problem $\mathcal{S}$-prodLH is either in P or is NP-complete. To the best of our knowledge, ours is the first systematic inquiry into the complexity of product state problems.

▶ **Theorem 1.** *For any fixed set of 2-qubit Hamiltonian terms $\mathcal{S}$, if every matrix in $\mathcal{S}$ is 1-local then $\mathcal{S}$-prodLH is in P, and otherwise $\mathcal{S}$-prodLH is NP-complete.*

[2] A more complex unpublished proof based on large graph cycles was known earlier by Wright [40].
[3] An earlier version of this work considered *exact* versions of product state and graph problems. We have now improved our hardness results to hold up to an inverse-polynomial additive gap.

Additionally, our hardness constructions only require coupling strengths (weights) of at most constant magnitude. This is preferable in practice and contrasts with most known QMA-hardness constructions.

The sets for which [14] shows $\mathcal{S}$-LH is in P are the same for which we show $\mathcal{S}$-PRODLH is in P, i.e. those containing only 1-local terms. This immediately implies that a family of 2-local Hamiltonians has efficiently-computable minimum product state energy if and only if it has efficiently-computable ground state energy.

▶ **Corollary 2.** *For any fixed set of 2-qubit Hamiltonian terms $\mathcal{S}$, the problem $\mathcal{S}$-LH is in* P *if and only if $\mathcal{S}$-PRODLH is in* P*.*

▶ **Corollary 3.** *For any fixed set of 2-qubit Hamiltonian terms $\mathcal{S}$, the problem $\mathcal{S}$-LH is* NP*-hard if and only if $\mathcal{S}$-PRODLH is* NP*-complete.*

Our results imply that hardness of product state approximations is not restricted to Hamiltonians for which product states well approximate ground states: for any QMA-hard family of terms, our result implies that (assuming QMA $\neq$ NP) we can construct a family of local Hamiltonians that are NP-hard to product state approximate *and* for which the product states do not well approximate the ground state by constructing Hamiltonians on two systems, each with one of these properties, and taking a disjoint union thereof. This implies that algorithms using product states to approximate the ground states of QMA-hard Hamiltonians face a "double penalty": hardness of approximating product states which themselves imperfectly approximate ground states.

The Stretched Linear Vector Max-Cut problem

Our hardness constructions for product state problems embed an objective function which we prove is NP-complete. This objective function generalizes the classical MAX-CUT problem. Given work on other variations of MAX-CUT, we expect this problem and our reductions should be of independent interest, especially to the optimization and approximation communities.

In MAX-CUT, one is given a graph $G = (V, E)$ and asked to assign to each vertex v a label $\hat{v} = \pm 1$ so as to achieve the maximum number of oppositely labeled adjacent vertices:

$$\mathrm{MC}(G) = \frac{1}{2} \max_{\hat{\imath}=\pm 1} \sum_{ij \in E} (1 - \hat{\imath}\hat{\jmath}) = \frac{1}{2} \max_{\hat{\imath}=\pm 1} \sum_{ij \in E} |\hat{\imath} - \hat{\jmath}|. \tag{1}$$

A problem referred to as VECTOR MAX-CUT, RANK-k-MAX-CUT, or MAX-CUT$_k$ (MC$_k$) has been studied [8, 7, 22] which generalizes MAX-CUT to assigning k-dimensional unit vectors so as to maximize the angles between adjacent vertex labels, or equivalently to maximize the squared distances between adjacent vertex labels:

$$\mathrm{MC}_k(G) = \frac{1}{2} \max_{\hat{\imath} \in S^{k-1}} \sum_{ij \in E} (1 - \hat{\imath} \cdot \hat{\jmath}) = \frac{1}{4} \max_{\hat{\imath} \in S^{k-1}} \sum_{ij \in E} \|\hat{\imath} - \hat{\jmath}\|^2. \tag{2}$$

Our new problem can be seen as a stretched and linear version of MC$_k$. The goal in W-LINEAR-MAX-CUT (MC$^{\mathrm{L}}_W$) is to assign unit vectors so as to maximize the distance between adjacent labels:

$$\mathrm{MC}^{\mathrm{L}}_{\mathrm{W}}(G) = \frac{1}{2} \max_{\hat{\imath} \in S^{d-1}} \sum_{ij \in E} \|W\hat{\imath} - W\hat{\jmath}\| = \frac{1}{2} \max_{\hat{\imath} \in S^{d-1}} \sum_{ij \in E} \sqrt{\|W\hat{\imath}\| + \|W\hat{\jmath}\| - 2(W\hat{\imath})^\top (W\hat{\jmath})}, \tag{3}$$

where W is a fixed $d \times d$ diagonal matrix. Comparing MC_k and $\mathrm{MC}^{\mathrm{L}}_W$, our problem sums over un-squared distances and incorporates a linear stretch given by W. We consider the decision version of this problem, in which the objective is to test whether the optimal solution is at least b or no more than a, for $b - a \geq 1/\operatorname{poly}(n)$. Note that, unlike $\mathcal{S}$-LH, this is an unweighted problem – one could naturally define a weighted version but our hardness results will not require this.

Geometrically, MC_k corresponds to embedding a graph into the surface of a unit sphere with the objective of maximizing the sum of the squared lengths of every edge. Likewise, our problem $\mathrm{MC}^{\mathrm{L}}_W$ corresponds to embedding a graph into the surface of a d-dimensional ellipsoid, with radii defined by the entries of W, with the objective of maximizing the sum of the (non-squared) edge lengths.

Despite being generalizations of the NP-complete Max-Cut problem, hardness of neither MC_k nor $\mathrm{MC}^{\mathrm{L}}_W$ is trivial. The Goemans-Williamson approximation algorithm for Max-Cut on an n-vertex graph begins with efficiently computing the solution to MC_n via an SDP. In fact, deciding MC_k is known to be in P for any $k = \Omega(\sqrt{|V|})$, [28, Theorem 8.4] or [3, (2.2)]. And while it has been conjectured by Lovász that MC_k is NP-complete for all constants $k \geq 1$, in [29, p. 236] and earlier, no proof has been given for any $k > 1$.

Our main theorem concerning W-linear-Max-Cut, which is used to prove Theorem 1, is the following.

▶ **Theorem 4.** *For any fixed non-negative $W = \operatorname{diag}(\alpha, \beta, \gamma)$ with at least one of α, β, γ nonzero, W-linear-Max-Cut is NP-complete.*

Quantum Max-Cut Product States and MC_3

As a corollary of our classification theorem, we give the first published proof of the fact that product state optimization in the Quantum Max-Cut (QMC) model is NP-hard. This model, also known as the anti-ferromagnetic Heisenberg model, is equivalent to $\mathcal{S}$-LH with $\mathcal{S} = \{XX + YY + ZZ\}$. We note that a sketch of a different proof for this specific problem was previously known but unpublished [40]. That proof was based on large graph cycles, and our gadgets are simpler to analyze.

However, the proof of Theorem 1 utilizes Hamiltonian gadgets involving negative weights (unlike the aforementioned proof of [40]). This leaves open whether prodQMC remains NP-hard on unweighted graphs. In Section 5, we give a direct proof of hardness using the fact that the unweighted product-state version of QMC is equivalent to MC_3 (Equation (2)). Our work then is also the first published proof that MC_k is NP-complete for some $k > 1$ (in our case, $k = 3$), partially resolving a conjecture of Lovász [29, p. 236]. Note that as with $\mathrm{MC}^{\mathrm{L}}_W$, we consider the decision version in which the goal is to determine whether the value is above b or below a, for b and a with inverse-poly separation.

▶ **Theorem 5.** MC_3 *is NP-complete.*

▶ **Corollary 6.** Quantum Max-Cut *restricted to product states,* prodQMC*, is NP-complete, even when all terms are restricted to have positive unit weight.*

1.2 Proof Overview

2-local $\mathcal{S}$-prodLH

As product states have classical descriptions and their energies can be calculated in polynomial time, $\mathcal{S}$-prodLH is automatically in NP, so we focus on how we show hardness. Our approach is in two parts. We show how to reduce $\mathrm{MC}^{\mathrm{L}}_W$ to $\mathcal{S}$-prodLH, and later we show that $\mathrm{MC}^{\mathrm{L}}_W$

NP-complete (Theorem 4). More precisely, the first part of our approach is to show that for any $\mathcal{S}$ containing a strictly 2-local term, there exists a corresponding weight matrix W meeting the conditions of Theorem 4 so that $\mathrm{MC}^{\mathrm{L}}_W$ is NP-hard. For any instance of $\mathrm{MC}^{\mathrm{L}}_W$ with this fixed W, we show how to construct a Hamiltonian from $\mathcal{S}$ such that the minimum product state energy encodes the $\mathrm{MC}^{\mathrm{L}}_W$ value, yielding Theorem 1.

We interpret problems over product states as optimization problems on the collection of Bloch vectors for each single-qubit state. For example, consider $\mathcal{S}$-PRODLH with the specific set $\mathcal{S} = \{XX + YY + ZZ\}$ (the QMC model). In this case, by writing each qubit v in the Bloch vector representation

$$|\phi_v\rangle\langle\phi_v| = \frac{1}{2}\left(I + v_1 X + v_2 Y + v_3 Z\right),$$

the energy contributed by an interaction between qubits u and v is

$$\mathrm{tr}\left(\left(X^u X^v + Y^u Y^v + Z^u Z^v\right)|\phi_u\rangle\langle\phi_u|\,|\phi_v\rangle\langle\phi_v|\right) = u_1 v_1 + u_2 v_2 + u_3 v_3. \tag{4}$$

So, given a Hamiltonian which is the sum of XYZ interactions between pairs of qubits, the problem of estimating the extremal product state energies is equivalent to optimizing the objective function

$$\sum_{uv} w_{uv}\, u \cdot v$$

over 3-dimensional unit vectors, where each edge uv corresponds to the Hamiltonian's (weighted) interaction graph. Up to constant shifts and scaling, this is equivalent to MC_3, introduced in Equation (2).

More generally, because the Pauli matrices are a basis for all Hermitian matrices, any 2-qubit interaction can be written as $H = \sum_{i,j=1}^{3} M_{ij}\sigma_i\sigma_j + \sum_{k=1}^{3}(*)c_k\sigma_k I + w_k I\sigma_k$, where the σ_i are the Pauli matrices, for some M, c, w. Then, the energy of a given product state is calculated similarly to Equation (4). However, the resulting expression potentially contains many terms.

Our approach is to take an arbitrary 2-qubit term and insert it into gadgets which simplify the energy calculations. First, in the proof of Theorem 1, we borrow a trick of [14] and symmetrize the terms. For any 2-qubit interaction H^{ab} on qubits a, b, the combined interaction $H^{ab} + H^{ba}$ is symmetric, invariant under swapping the qubits. A similar trick handles the case of anti-symmetric terms.

Second, in the proofs of Lemmas 14 and 15, we show how to use a symmetric or anti-symmetric term to embed the $\mathrm{MC}^{\mathrm{L}}_W$ value into the minimum energy of a gadget. We begin by removing the 1-local terms, such as $\sigma_1 I$ or $I\sigma_2$, again taking inspiration from gadgets used by [14]. For two qubits u, v corresponding to two vertices in a $\mathrm{MC}^{\mathrm{L}}_W$ instance, the gadget adds two ancilla qubits and weights each interaction within the gadget to effectively cancel out the 1-local terms. When the two ancilla qubits vary freely, we find the minimum energy contributed by the entire four-qubit gadget is determined by the distance between the states of u and v. Although each individual edge contributes energy proportional to the squared distance between their states, the overall gadget contributes energy proportional to just the distance of the two "vertex qubits", $-\|Mu - Mv\|$. With some massaging, we can treat M as a non-negative diagonal matrix which meets the conditions of Theorem 4.

Therefore, as desired, we have that the $\mathrm{MC}^{\mathrm{L}}_W$ value for an NP-complete instance of $\mathrm{MC}^{\mathrm{L}}_W$ can be embedded into the minimum product state energy of an $\mathcal{S}$-PRODLH instance.

To prove that $\mathrm{MC}^{\mathrm{L}}_W$ is NP-hard for any fixed diagonal non-negative W with at least one nonzero entry, we divide into three cases, in Lemmas 21–23. Our first proof is a reduction from the standard Max-Cut problem, while the other two are reductions from 3-Coloring.

When there is a unique largest entry of W, we reduce from Max-Cut by taking any input graph G and forming G' by adding large star gadgets around each of the vertices of G, each using many ancilla vertices. Because the ancilla vertices in each gadget have just one neighbor (the original vertex at the center of the gadget), their optimal vector labels given any choice of labels for the center vertices are the negation of the center vertex labels. This means they heavily penalize assigning the center vertices any labels that are not along the highest-weight axis. Therefore, when the maximum entry of W is unique, the optimal $\mathrm{MC}^{\mathrm{L}}_W$ assignment to G' will have its vector labels almost entirely along the highest-weight axis. The assignment can trivially earn the maximum possible value on the star gadgets, and the amount additional amount it can earn on the original edges of G corresponds to the Max-Cut value of G.

When all of the entries of W are equal, we reduce from 3-Coloring. Given a graph G, we construct G' by replacing each edge with a 4-clique gadget, made by adding one ancilla vertex per gadget, along with a single ancilla vertex shared by every gadget. We show that G is 3-colorable iff G' has a sufficiently large $\mathrm{MC}^{\mathrm{L}}_W$ value. Specifically, we show this holds iff there is a vector assignment that simultaneously achieves (nearly) the maximum value on all of the clique gadgets. Achieving the maximum objective value on a clique corresponds to maximizing the total distance between each pair of vectors, and this enforces a predictable arrangement.

When weights are equal, assigning these vectors can be viewed as inscribing vectors in the unit sphere, and it is known that maximal perimeter polyhedra inscribed in the sphere must be regular. So for a 4-clique, the vector labels must form a regular tetrahedron. We carefully argue that for regular tetrahedra, fixing two of the vertices (approximately) fixes the other two vertices up to swapping (several of these geometric facts are proved in Section A). This means that the clique gadget corresponding to an edge uv in G shares two vertices with any gadget corresponding to an edge vw incident to the first edge: v and the "global" ancilla shared by all gadgets. This means that, once we fix the vector assigned to the global ancilla, the choice of a vector for v restricts the labels of both u and w to be chosen from a set of two vectors. So simultaneously optimizing every clique gadget is possible iff, for each connected component of G', we can 3-color that component with three vectors (corresponding to, for any v in the component, the vector assigned to v, the vector assigned to the global ancilla vertex, and the two vectors that can share a maximal tetrahedron with those two).

Finally, when the two largest entries of W are equal but distinct from the third, we combine the two previous approaches. Inserting star gadgets effectively reduces the problem from three dimensions to two, by penalizing vector assignments not in the 2d space corresponding to the two largest entries of W. We then add 3-clique gadgets, with one ancilla for each edge in G, and optimizing these over two dimensions corresponds to inscribing maximal perimeter triangles in the unit circle. Now, assigning a vector to one vertex fixes the optimal vectors assigned to the other two vertices (again up to swapping them) and so there is again a one-to-one correspondence between vector assignments simultaneously optimizing every clique gadget and 3-colorings of the connected components of G.

Three-Dimensional Vector Max-Cut

Our proof that MC_k with positive unit weights is NP-complete runs on very similar lines to our 3-Coloring reduction for $\mathrm{MC}^{\mathrm{L}}_W$ with all weights equal. However, there is one additional complication: that proof depended on the fact that the sum of the side lengths of the tetrahedron is uniquely maximized by choosing it to be regular. This is not the case when we instead consider the squared side lengths, for instance assigning half the vectors to one pole of the sphere and half to the other would achieve the same bound. So we use a different gadget: instead of replacing every edge with a 4-clique, we replace every edge with a 4-clique that in turn has its edges replaced with triangles. It turns out that this gadget *does* have a unique optimal MC_k assignment, which in particular assigns the vectors in the 4-clique to a regular tetrahedron, allowing us to proceed along the same lines as the aforementioned proof.

1.3 Related Work

Brandão and Harrow [4] give simple conditions under which 2-local Hamiltonians have product states achieving near-optimal energy, such as systems with high-degree interaction graphs. This suggests that unless NP = QMA, such Hamiltonians cannot be QMA-complete. Since then, there has been a line of work on the relationship between product states and general ground states in other Hamiltonians, in the more general case when the two problems are not equivalent. See e.g. [5, 17, 20].

Product states have especially been studied in the context of the Quantum Max-Cut problem, introduced by Gharibian and Parekh [18]. Briët, de Oliveira Filho, and Vallentin [8] give hardness results conditional on the unique games conjecture for approximating the optimal product state in the QMC model. Related work by Hwang, Neeman, Parekh, Thompson, and Wright [22] also gives tight hardness results for the QMC problem under a plausible conjecture. Parekh and Thompson [32] give an optimal approximation algorithm for QMC when using product states.

See [22] for an exposition of the relationship between QMC restricted to product state solutions and the Vector Max-Cut problem. Studying vector solutions to Max-Cut has a long history [29], including the seminal Goemans-Williamson algorithm [19]. This study is usually with the goal of a solution to the original Max-Cut problem, which relates to approximation ratios of integer and semidefinite programs. Bounding these ratios has been referred to in terms of Grothendieck problems and inequalities: see [7, 2] for further context on this nomenclature. A tight NP-hardness result is known for the non-commutative Grothendieck problem [9] which also generalizes the "little" Grothendieck problem over orthogonal groups [2]. Iterative algorithms (heuristics) for solving MC_k are also well-studied in the literature (see [11] and citing references), since in practice solving MC_k is often faster than solving the corresponding SDP relaxation.

Cubitt and Montanaro [14] classified $\mathcal{S}$-LH for sets $\mathcal{S}$ of 2-qubit terms. This work relies in turn on the work of Bravyi and Hastings [6] to classify the StoqMA case. [14] also examined a variant where $\mathcal{S}$ is assumed to always contain arbitrary single-qubit terms. Follow-up work by Piddock and Montanaro [33] began investigating classifying the complexity of $\mathcal{S}$-LH under the additional restrictions of positive weights (anti-ferromagnetic model) and/or interactions restricted to a spatial geometry such as a 2D grid. [15, 34] continued along these lines, and introduced Hamiltonian simulations rather than computational reductions.

In classical computer science, Schaefer [35] gave a dichotomy theorem showing that given a fixed set of allowed constraints, the family of CSPs is either decidable in P or is NP-complete; but see Section 2 or [14] for some important assumptions that are made in the

quantum versus classical models. In fact, Schaefer's classification offers a more fine-grained classification with classes within P. Later, [12, 25] gave a similar dichotomy theorem for the complexity of Max-SAT and related optimization problems, where the question is not just whether all clauses are simultaneously satisfiable but how many are simultaneously satisfiable. Applying weights to constraints becomes relevant with Max-SAT, and is covered in their work. This work is especially relevant given k-LH is more analogous to Max-SAT than to SAT. Continuing, Jonsson [23] classified these problems when both positive and negative weights are allowed. While arbitrary weights seem natural in the quantum setting, previous classical work simply assumed all weights were non-negative. The book [13] offers an excellent survey of this area. The more recent results of [24, 36] extend classical classification results to problems with non-binary variables (analogous to qu*d*its).

1.4 Open Questions

1. We have shown a relationship between when product state problems and general Hamiltonian problems are hard. This points toward an important question: can some "interesting" class of local Hamiltonians or a Hamiltonian problem for which we do not know an explicit efficient algorithm be proven *not* hard, e.g. neither NP- nor QMA-hard, by showing the corresponding product state problem is in P?
2. Is a more refined classification of the complexity of product state problems, taking into account allowable weights or spatial geometry in the vein of [33], or imposing other promises, possible?
3. While little progress has been made classifying the general $\mathcal{S}$-LH problem for higher-locality families, can we classify $\mathcal{S}$-prodLH for families of k-local Hamiltonians with $k > 2$?
4. Can we relate approximability instead of just complexity? For example, does the ability to α-approximate the product state problem imply the ability to β-approximate the general ground state problem on families defined by some sets of allowed interactions but not others?
5. As mentioned above, we make the first progress towards a conjecture of Lovász [29] that MC_k is NP-hard for any $k = O(1)$. We only focus on $k = 3$ here because of our interest in prodLH. Can our proof be generalized to other values of k?

2 Preliminaries

We assume familiarity with the conventions of quantum computation [38] and complexity theory [1, 26]. See also [16, 14] for surveys of Hamiltonian complexity.

2.1 Notation

I denotes the identity operator. $\lambda_{\min}(H)$ and $\lambda_{\max}(H)$ denote the minimum and maximum eigenvalues of an operator H. In the same manner as with asymptotic $\mathrm{O}(\cdot)$ notation, we use $\mathrm{poly}(n)$ to denote a term that can be bounded by some fixed polynomial in n.

For an operator A, we use superscripts such as A^{abc} to indicate A acts on individual qubits a, b, and c. Unless A is symmetric, the order matters and A^{ab} is different than A^{ba}. If no superscripts are used, then the action is implicit in the ordering of the terms (left versus right).

When clear from context, we will denote the tensor product of two operators $A \otimes B$ simply by AB. All terms implicitly are tensor the identity on any systems not specified.

SWAP will denote the 2-qubit operator exchanging $|01\rangle$ and $|10\rangle$ while $|00\rangle$ and $|11\rangle$ unchanged. We call a 2-qubit term H *symmetric* if $H = \mathrm{SWAP}(H)\,\mathrm{SWAP}$, meaning the ordering of the qubits does not matter. Alternatively, H is *antisymmetric* if $H = -\mathrm{SWAP}(H)\,\mathrm{SWAP}$.

The single-qubit Pauli matrices are denoted X, Y, Z or $\sigma_1, \sigma_2, \sigma_3$. Recall that $\{X, Y, Z, I\}$ is a basis for 2×2 Hermitian matrices. The *Pauli decomposition* of a 2-qubit Hermitian matrix H is H written in the Pauli basis,

$$H = \sum_{i,j=1}^{3} M_{ij}\sigma_i\sigma_j + \sum_{k=1}^{3}(v_k\sigma_k I + w_k I\sigma_k), \tag{5}$$

with all coefficients real and the 3×3 matrix M referred to as the *correlation matrix*. Generally, Equation (5) should include a term wII, but we will work with traceless terms such that $w = 0$.

Unless otherwise stated, all graphs are undirected and simple, meaning there are no self-loops and no multi-edges. We assume all graphs are connected, as it is straightforward to extend any of our constructions to disconnected graphs. When summing over edges, $\sum_{ij\in E}$, we do not double-count ij and ji. Finally, $S^i = \{x \in \mathbb{R}^{i+1} : \|x\| = 1\}$ denotes the unit sphere in $(i+1)$-dimensional space.

2.2 Definitions and Assumptions

A k-local Hamiltonian on n qubits is a Hermitian matrix $H \in \mathbb{R}^{2^n \times 2^n}$ that can be written as $H = \sum_{i=1}^{m} H_i$ such that each H_i is Hermitian and acts non-trivially on at most k qubits. More precisely, each H_i acts on some subset S_i of at most k qubits and each term in the sum is $H_i \otimes I^{[n]\setminus S_i}$, but we generally leave this implicit. We usually consider constant values of k, so each term is of constant size independent of n. The k-qubit terms H_i are often referred to as *interactions* between qubits. We may refer to eigenvalues and expectation values $\langle\psi| H |\psi\rangle$ as the *energy* of the state $|\psi\rangle$ in the system described by H. In particular, the *ground state energy* and *ground state* refer to the minimum eigenvalue and an associated eigenvector.

Estimating the minimum eigenvalue of a Hamiltonian is a natural quantum generalization of estimating the maximum number of satisfiable clauses in a Boolean formula.

▶ **Definition 7** (k-LH). *Given a k-local Hamiltonian $H = \sum_{i=1}^{m} H_i$ acting on n qubits with $m = \mathrm{poly}(n)$, the entries of each H_i specified by at most $\mathrm{poly}(n)$ bits, and the norms $\|H_i\|$ polynomially bounded in n, and two real parameters b, a such that $b - a \geq 1/\mathrm{poly}(n)$, decide whether $\lambda_{\min}(H)$ is at most a (YES) or at least b (NO), promised that one is the case.*

In this work, we are interested in k-LH restricted to families of local Hamiltonians, where the families are determined by sets of allowed interactions. In particular, we will be interested in sets of 2-qubit interactions.

▶ **Definition 8** ($\mathcal{S}$-LH). *For $\mathcal{S}$ any fixed set of Hamiltonian terms, define $\mathcal{S}$-LH as the problem k-LH with the additional promise that any input is of the form $\sum_{i=1}^{m} w_i H_i$ where each H_i is an element of $\mathcal{S}$ assigned to act on some subset of qubits and the* weights $w_i \in \mathbb{R}$ *have magnitude polynomially-bounded in n.*

▶ Remark 9. There are several standard assumptions implicit in our definition of $\mathcal{S}$-LH. Some are not physically realistic in the context of the condensed-matter literature but allow us to precisely characterize the complexity of these problems. First, although classical CSPs generally allow a constraint to take as input multiple copies of the same variable, this makes

less sense in the quantum setting and we do not allow it. Second, the definition of k-local only restricts the dimension of each term, it does not imply any spatial locality or geometry. Therefore, any term in H may be applied to any subset of qubits with the qubits arranged in any order. In particular this means that, if $\mathcal{S}$ contains a directed term H^{ab}, then the family of Hamiltonians allowed as input to $\mathcal{S}$-LH is equivalent to the family allowed to $\mathcal{S}'$-LH for $\mathcal{S}' = \mathcal{S} \cup \{H^{ba}\}$. Third, for the purpose of classifying the complexity of $\mathcal{S}$-LH, we may assume $I \in \mathcal{S}$, since adding or removing a term wI is equivalent to simply shifting the input parameters a, b by w. For $\mathcal{S}$ containing 2-qubit terms, this fact also implies we may assume all terms in $\mathcal{S}$ are traceless. Fourth, except when noted, we allow both positive and negative weights.

Classifying the complexity of systems under additional, more physically natural restrictions appears to be a significantly more difficult problem [14, 33].

Given this setup, our interest will be in the problems k-LH and $\mathcal{S}$-LH restricted to product states.

▶ **Definition 10** (Product state). *A state $\rho = \bigotimes_{i=1}^{n} \rho^i$ where each ρ^i is a single-qubit state.*

▶ **Definition 11** (PRODLH). *Given a k-local Hamiltonian $H = \sum_{i=1}^{m} H_i$ on n qubits with $m = \text{poly}(n)$, the entries of each H_i specified by at most $\text{poly}(n)$ bits, and the norms $\|H_i\|$ polynomially bounded in n, and two real parameters $b \geq a$, decide whether there exists a product state ρ with $\text{tr}(\rho H) \leq a$ (YES) or all product states satisfy $\text{tr}(\rho H) \geq b$ (NO), promised that one is the case.*

The problem $\mathcal{S}$-PRODLH is defined analogously. In both definitions, the fact that product states have concise classical descriptions allows us to naturally consider any choice of parameters, even an exact decision problem with $b = a$, in contrast to k-LH.

By convexity, a product state ρ achieves an extreme value of $\text{tr}(\rho H)$ if and only if there exists a pure product state $|\psi\rangle$ which achieves that value. Similarly, mixtures of product states, known as separable states, reduce to considering pure product states.

▶ Remark 12. In the context of $\mathcal{S}$-LH or $\mathcal{S}$-PRODLH given some fixed set S, we will describe $\mathcal{S}$ as "containing" a Hamiltonian term H, or that we "have access to" H, even when formally $H \notin \mathcal{S}$. As previously referenced in Remark 9, given a set $\mathcal{S}$, the family of Hamiltonians allowed as input to $\mathcal{S}$-LH may be equivalent to the family allowed given some other set $\mathcal{S}'$. For example, $\mathcal{S}'$ may include $\{PHP^\dagger\}$ for $H \in \mathcal{S}$ and any permutation of the qubits P. Similarly, adding constant multiples of the terms in $\mathcal{S}$ or any linear combinations of terms from $\mathcal{S}$ does not change the family of allowed Hamiltonians. So, when discussing $\mathcal{S}$-LH, we may implicitly refer to elements of the largest set $\mathcal{S}'$ such that $\mathcal{S}$-LH and $\mathcal{S}'$-LH each have the same family of allowed inputs.

Additionally, we note that $\mathcal{S}$-LH is reducible to $\mathcal{S}'$-LH for any $\mathcal{S}'$ which can be used to implement all elements of $\mathcal{S}$ – whether because formally $\mathcal{S} \subseteq \mathcal{S}'$ or through other means. In the opposite direction, if the terms in a set $\mathcal{S}$ can be used to construct some term H and we wish to show hardness of $\mathcal{S}$-LH, then it is sufficient to show hardness of $\{H\}$-LH.

Finally, for a 2-local Hamiltonian, we may refer to the *interaction graph*, with vertices associated with each qubit such that vertex i is adjacent to vertex j whenever a nonzero interaction exists on qubits i and j. When all interactions are symmetric, then the graph is undirected. Notably, when $\mathcal{S}$-LH is defined with $\mathcal{S}$ a singleton, then an input is fully specified by its weighted interaction graph.

3 Classification of $\mathcal{S}$-prodLH

In this section, we prove a dichotomy theorem classifying the complexity of estimating the minimum expectation of product states for given families of 2-local Hamiltonians. In particular, we show that for any set $\mathcal{S}$ of 2-qubit terms such that at least one term is not 1-local,[4] the problem $\mathcal{S}$-prodLH is NP-complete. These are precisely those 2-local families such that (as shown in [14]) $\mathcal{S}$-LH is QMA-, StoqMA-, or NP-complete. Conversely, if all terms are 1-local, then both $\mathcal{S}$-LH and $\mathcal{S}$-prodLH are in P.

▶ **Theorem 1.** *For any fixed set of 2-qubit Hamiltonian terms $\mathcal{S}$, if every matrix in $\mathcal{S}$ is 1-local then $\mathcal{S}$-prodLH is in P, and otherwise $\mathcal{S}$-prodLH is NP-complete.*

Our NP-hardness results hold with coupling strengths of at most constant magnitude.

For comparison with our classification, we recall the tetrachotomy theorem of Cubitt and Montanaro [14] classifying $\mathcal{S}$-LH for families of 2-local Hamiltonians. They proved that for every set of 2-qubit Hamiltonian terms $\mathcal{S}$, the problem $\mathcal{S}$-LH is either in P or NP-, StoqMA-, or QMA-complete, and described properties of the set $\mathcal{S}$ which determine the problem's complexity. We note that both Theorem 13 and our Theorem 1 classify the complexity of *all* sets of 2-qubit terms.

▶ **Theorem 13** (Theorem 7 of [14])**.** *For $\mathcal{S}$ any fixed set of 2-qubit Hamiltonian terms:*

- *If every matrix in $\mathcal{S}$ is 1-local, then $\mathcal{S}$-LH is in P.*
- *Otherwise, if there exists a single-qubit unitary U such that U locally diagonalizes all elements of $\mathcal{S}$ (i.e. $U^{\otimes 2} H U^{\dagger \otimes 2}$ is diagonal for each 2-qubit $H \in \mathcal{S}$), then $\mathcal{S}$-LH is NP-complete.*
- *Otherwise, if there exists a single-qubit unitary U such that for each 2-qubit $H \in \mathcal{S}$,*

$$U^{\otimes 2} H U^{\dagger \otimes 2} = \alpha Z^{\otimes 2} + A \otimes I + I \otimes B$$

for some $\alpha \in \mathbb{R}$ and 1-local Hermitian matrices A, B, then $\mathcal{S}$-LH is StoqMA-complete.

- *Otherwise, $\mathcal{S}$-LH is QMA-complete.*

Combining our classification of $\mathcal{S}$-prodLH with the classification of $\mathcal{S}$-LH gives us Corollaries 2 and 3.

To prove Theorem 1, showing containment in P and NP are straightforward, and our effort is to prove NP-hardness. In the proof, we will use a simple symmetrization trick that allows us to consider only antisymmetric or symmetric Hamiltonian terms. We then prove two lemmas, one for each case.

▶ **Lemma 14.** *If $\mathcal{S}$ contains a 2-qubit antisymmetric term that is not 1-local, then $\mathrm{MC}^{\mathrm{L}}_W$ with $W = \mathrm{diag}(1, 1, 0)$ is polynomial-time reducible to $\mathcal{S}$-prodLH.*

▶ **Lemma 15.** *If $\mathcal{S}$ contains a 2-qubit symmetric term that is not 1-local, then there exists a fixed non-negative $W = \mathrm{diag}(\alpha, \beta, \gamma)$ with at least one of α, β, γ nonzero such that $\mathrm{MC}^{\mathrm{L}}_W$ is polynomial-time reducible to $\mathcal{S}$-prodLH.*

In Section 4 we prove Theorem 4, that $\mathrm{MC}^{\mathrm{L}}_W$ is NP-complete for any $W = \mathrm{diag}(\alpha, \beta, \gamma)$ that is nonzero and non-negative.

We state some helpful facts in Section 3.1 below and then prove the two lemmas in Section 3.2. We will now use these lemmas to prove our main theorem.

[4] We would prefer a more concise name for 2-qubit terms that are not 1-local, but are unaware of any. One option is 2-qubit terms with *Pauli degree* 2. Alternatively, these are 2-qubit terms which have nonzero *Pauli rank*, referring to the rank of the correlation matrix M in the Pauli decomposition (Equation (5)).

Proof of Theorem 1. First consider the case where $\mathcal{S}$ only contains 1-local terms. Then we can write $H = \sum_i H_i$, where H_i acts only on the i^{th} qubit. If $|\psi_i\rangle$ is the single-qubit state minimizing $\langle\psi_i|H_i|\psi_i\rangle$, then $|\psi\rangle = \bigotimes_i |\psi_i\rangle$ minimizes $\langle\psi|H|\psi\rangle$, and so $\mathcal{S}$-prodLH can be solved by finding the ground state of n single-qubit Hamiltonians, which takes $O(n)$ time.

Now suppose $\mathcal{S}$ is a set of of 2-qubit Hamiltonian terms such that at least one element of $\mathcal{S}$ is not 1-local. Let H be any such element. As previously mentioned, $\mathcal{S}$-prodLH $\in$ NP for any fixed $\mathcal{S}$, as product states have concise classical descriptions which can be used to efficiently calculate expectation values for a given local Hamiltonian. If H is antisymmetric, then MC^{L}_W with fixed $W = \text{diag}(1, 1, 0)$ is reducible to $\mathcal{S}$-prodLH by Lemma 14, and MC^{L}_W with such a W is NP-hard by Theorem 4, so $\mathcal{S}$-prodLH is NP-complete. If H is symmetric, then by Lemma 15 there exists a non-negative nonzero matrix $W = \text{diag}(\alpha, \beta, \gamma)$ such that MC^{L}_W is reducible to $\mathcal{S}$-prodLH, it is NP-hard by Theorem 4, and so $\mathcal{S}$-prodLH is NP-complete. If H is neither of these, then we use our freedom to permute the direction H is applied to any pair of qubits a, b to apply both H^{ab} and H^{ba}, which is equivalent to implementing the symmetric term $H' = H + \text{SWAP}\,(H)\,\text{SWAP}$. So, we can say $\mathcal{S}$ effectively contains H', or formally that hardness of $\{H'\}$-prodLH implies hardness of $\mathcal{S}$-prodLH, and again referring to Lemma 15 concludes the proof. ◀

3.1 Closure Properties of $\mathcal{S}$-prodLH

Before proving the two lemmas required in the proof of Theorem 1, we review several more facts regarding 2-qubit Hamiltonian terms and operations under which the complexities of $\mathcal{S}$-LH and $\mathcal{S}$-prodLH are unaffected. This section mostly reviews observations made in [14].

First, for a single-qubit unitary U and an operator H, define *simultaneous conjugation* by U to mean $U^{\otimes n} H U^{\dagger \otimes n}$. When discussing sets $\mathcal{S}$ of k-qubit terms, we define simultaneous conjugation to mean $\{U^{\otimes k} H U^{\dagger \otimes k} | H \in \mathcal{S}\}$.

▶ **Fact 16.** *For any single-qubit unitary U, the complexities of $\mathcal{S}$-LH and $\mathcal{S}$-prodLH are equal to the complexities of $\mathcal{S}'$-LH and $\mathcal{S}'$-prodLH, respectively, where $\mathcal{S}'$ is $\mathcal{S}$ simultaneously conjugated by U.*

Observe that $U^{\otimes n} \left(\sum_{i=1}^m H_i\right) U^{\dagger \otimes n} = \sum_{i=1}^m U^{\otimes k} H_i U^{\dagger \otimes k}$. Simultaneous conjugation by U gives a bijection between Hamiltonians allowed in $\mathcal{S}$-LH and $\mathcal{S}'$-LH as well as $\mathcal{S}$-prodLH and $\mathcal{S}'$-prodLH. The above fact follows from observing that this bijection preserves expectation values, and that $U^{\otimes n} |\phi\rangle$ is a product state iff $|\phi\rangle$ is.

As an application of Fact 16, the following is based on an observation in [14].

▶ **Fact 17.** *For any choice of permutation π on $\{1, 2, 3\}$ and any choice of* two *of $c_1, c_2, c_3 = \pm 1$, there exists a single-qubit unitary U and corresponding third coefficient s.t. simultaneous conjugation by U maps the Pauli matrices $\{\sigma_1, \sigma_2, \sigma_3\}$ to $\{c_{\pi(1)}\sigma_{\pi(1)}, c_{\pi(2)}\sigma_{\pi(2)}, c_{\pi(3)}\sigma_{\pi(3)}\}$. So, writing every element of $\mathcal{S}$ in the Pauli basis, relabeling all σ_i with $c_{\pi(i)}\sigma_{\pi(i)}$ in the decompositions of each element of $\mathcal{S}$ does not change the complexity of $\mathcal{S}$-LH or $\mathcal{S}$-prodLH, where π and two of the coefficients can be chosen arbitrarily.*

To justify the above fact, consider simultaneously rotating the three axes of the Bloch sphere. Next, we quote the following, more involved, fact without proof.

▶ **Fact 18** ([14, 21]). *Let H be a 2-qubit Hamiltonian term with Pauli decomposition $H = \sum_{i,j=1}^{3} M_{ij}\sigma_i\sigma_j + \sum_{k=1}^{3}(v_k\sigma_k I + w_k I\sigma_k)$. For any single-qubit unitary U,*

$$U^{\otimes 2}H(U^\dagger)^{\otimes 2} = \sum_{i,j=1}^{3}(RMR^T)_{ij}\sigma_i\sigma_j + \sum_{k=1}^{3}((Rv)_k\sigma_k I + (Rw)_k I\sigma_k) \tag{6}$$

for some $R \in SO(3)$. Likewise, for any $R \in SO(3)$, there exists a single-qubit U such that the Pauli decomposition of $U^{\otimes 2}H(U^\dagger)^{\otimes 2}$ matches Equation (6).

A further straightforward observation from [14] is that in the Pauli decomposition (Equation (5)), if H is symmetric then the correlation matrix M is symmetric, and if H is antisymmetric then M is skewsymmetric, meaning $M = -M^\top$.

Finally, the below observation combines some of the above facts to establish a "normal form" for symmetric and antisymmetric terms.

▶ **Fact 19.** *If a 2-qubit Hamiltonian term H is symmetric, and so the associated correlation matrix M is symmetric, there exists $R \in SO(3)$ which diagonalizes M. Combining Facts 16 and 18, for any 2-qubit symmetric term H, there exists a symmetric term of the form $H' = \sum_{i=1}^{3} u_i\sigma_i\sigma_i + \sum_{j=1}^{3} v_j(\sigma_j I + I\sigma_j)$ such that the complexities of $\{H\}$-LH and $\{H\}$-prodLH are respectively the same as $\{H'\}$-LH and $\{H'\}$-prodLH.*

If H is a 2-qubit antisymmetric term that is not 1-local, then M is skewsymmetric and nonzero. Such an M may be block diagonalized via some $R \in SO(3)$ such that H is mapped to $a(\sigma_i\sigma_j - \sigma_j\sigma_i) + \sum_{k=1}^{3} v_k(\sigma_k I - I\sigma_k)$ [37, 14]. In particular, by Fact 17, H can be mapped to $a(XZ - ZX) + \sum_{k=1}^{3} v'_k(\sigma_k I - I\sigma_k)$. Therefore, the complexities of $\{H\}$-LH and $\{H\}$-prodLH are unaffected by assuming H has this form.

3.2 Proofs of Antisymmetric and Symmetric Lemmas

We now prove the two lemmas required in the proof of the main theorem, respectively handling the cases that $\mathcal{S}$ contains an antisymmetric term and that $\mathcal{S}$ contains a symmetric term. In both cases, it is sufficient for $\mathcal{S}$ to contain just a single term. Interestingly, our construction in Lemma 14 for antisymmetric terms is unweighted, meaning all weights are $+1$. In this case, the final Hamiltonian is fully determined by the specification of a single 2-qubit term and the interaction graph. Our construction in Lemma 15 uses positive and negative unit weights, ± 1. Intuitively, antisymmetric terms inherently allow negativity by simply permuting the qubits they act on, while for symmetric terms we must use negative weights.

Proof of Lemma 14. Consider an arbitrary instance of the problem $\mathrm{MC}^{\mathrm{L}}_W$ with $W = \mathrm{diag}(1, 1, 0)$. For a given graph $G = (V, E)$, the objective function is

$$\mathrm{MC}^{\mathrm{L}}_{\mathrm{W}}(G) = \frac{1}{2}\max_{i\in S^2}\sum_{ij\in E}\|W\hat{\imath} - W\hat{\jmath}\|.$$

Given a parameter C, we must decide whether $\mathrm{MC}^{\mathrm{L}}_{\mathrm{W}}(G)$ is at least C or at most $C - \varepsilon$. To reduce $\mathrm{MC}^{\mathrm{L}}_W$ to $\mathcal{S}$-prodLH, we first construct a gadget using the promised antisymmetric term. Then, we apply this gadget according to the graph G such that the minimum energy of a product state in our final Hamiltonian will equal $-\mathrm{MC}^{\mathrm{L}}_{\mathrm{W}}(G)$.

Denote the assumed 2-qubit antisymmetric term that is not 1-local in $\mathcal{S}$ by H. By Fact 18, our antisymmetric term H may be mapped to a term of the form

$$w\left(X^aZ^b - Z^aX^b\right) + \sum_{k=1}^{3} v_k\left(\sigma_k^a I^b - I^a\sigma_k^b\right)$$

where all coefficients are real, and we have $w \neq 0$ since the term is not 1-local. As explained in Remark 12 and Section 3.1, the complexity of $\{H\}$-PRODLH is equivalent to that of $\{H'\}$-PRODLH for H' derived using a variety of operations, including permutations and linear combinations. If w is negative, then we redefine the direction the term acts in, H^{ab} versus H^{ba}, so that w is positive. Finally, we scale[5] the term so that $w = 1$ and define a single-qubit Hermitian matrix $A = \sum_{k=1}^{3} v_k \sigma_k$. Given the complexity is unchanged using H or H', we simply redefine the original term, so that

$$H^{ab} = X^a Z^b - Z^a X^b + A^a I^b - I^a A^b.$$

Next, we use a symmetrization gadget to remove the 1-local terms $AI - IA$. For four qubits a, b, c, d, define

$$B = H^{ab} + H^{bc} + H^{cd} + H^{da}.$$

Note that here the direction of the interaction matters, since the terms are asymmetric. Then

$$B = (X^a - X^c)(Z^b - Z^d) - (Z^a - Z^c)(X^b - X^d).$$

Now we consider how B interacts with product states on four qubits. For $e = a, b, c, d$, define

$$\rho^e = \frac{1}{2}(I + r^e \cdot v^e)$$

with $v^e = (X^e, Y^e, Z^e)$ the Pauli operators and $r^e = (x^e, y^e, z^e)$ the Bloch vector. Then, writing any product state on qubits a, b, c, d as $\rho^a \rho^b \rho^c \rho^d$, the expectation value on B is $\operatorname{tr}\left(B \rho^a \rho^b \rho^c \rho^d\right)$. After eliminating terms, we find this equals

$$(r^a - r^c)^\top W' (r^b - r^d) \quad \text{for} \quad W' = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}.$$

It is helpful to note that $W'^\top = -W'$.

Now, consider the graph G given as input to $\mathrm{MC}^{\mathrm{L}}_W$. Associate a qubit with each vertex and call these the "vertex qubits". For each edge ij, construct a copy of B such that it acts on qubits ij and two ancilla qubits. The vertex qubits may be shared among several gadgets, while the ancilla qubits are part of only one gadget. In particular, we choose to associate the vertex qubits with qubits a and c in each copy of B, letting b and d be the ancilla. We will refer to the copy of B which acts on vertex qubits i and j as B^{ij}. Our Hamiltonian is then

$$H_{\text{final}} = \sum_{ij \in E} B^{ij}.$$

Before analyzing the full Hamiltonian H_{final}, consider the minimum expectation of a single gadget B^{ij} if the two vertex qubits are fixed, i.e. $\min_{r^b, r^d} (r^i - r^j)^\top W' (r^b - r^d)$. The minimum is achieved when $r^b = -W'^\top (r^i - r^j) / \left\| W'^\top (r^i - r^j) \right\|$ and $r^d = -r^b$, which yields an expectation of

$$-2 \left\| W'' (r^i - r^j) \right\| \quad \text{for} \quad W'' = \operatorname{diag}(1, 0, 1).$$

[5] If we want the terms to have unit weights, we could forgo scaling the term and reduce to $w\,\mathrm{MC}^{\mathrm{L}}_{\mathrm{W}}$ instead. As $w > 0$ this problem has the same complexity as $\mathrm{MC}^{\mathrm{L}}_{\mathrm{W}}$.

Therefore, given an arbitrary graph $G = (V, E)$, applying our gadget to every edge constructs a Hamiltonian such that the minimum expectation of any product state is equal to

$$2 \min_{r^i \in S^2} \sum_{ij \in E} -\left\|W''(r^i - r^j)\right\| = -2 \max_{r^i \in S^2} \sum_{ij \in E} \left\|W''(r^i - r^j)\right\|.$$

For any graph G, the objective $\mathrm{MC}^{\mathrm{L}}_{\mathrm{W}}(G)$ is equal for $W = \mathrm{diag}(1, 1, 0)$ and $W'' = \mathrm{diag}(1, 0, 1)$. Alternatively, we may use our freedom to relabel Paulis to redefine the 2-qubit term H such that the final weight matrix is W.

Finally, multiplying the full Hamiltonian H_{final} by $\frac{1}{2}$ gives us that the minimum expectation of any product state equals

$$-\max_{r^i \in S^2} \sum_{ij \in E} \left\|W(r^i - r^j)\right\| = -\,\mathrm{MC}^{\mathrm{L}}_{\mathrm{W}}(G).$$

We conclude deciding $\mathrm{MC}^{\mathrm{L}}_{W}$ reduces to deciding PRODLH on H_{final}. Since H_{final} is entirely constructed from the antisymmetric term $H \in \mathcal{S}$, this completes the desired reduction of $\mathrm{MC}^{\mathrm{L}}_{W}$ with $W = \mathrm{diag}(1, 1, 0)$ to $\mathcal{S}$-PRODLH. ◀

Next we prove the lemma dealing with $\mathcal{S}$ containing a symmetric term. The construction is nearly the same as the in the previous proof, but requires negative weights to implement the symmetrization gadget removing 1-local terms.

Proof of Lemma 15. Given fixed $\mathcal{S}$, we will show there exists some fixed W such that $\mathrm{MC}^{\mathrm{L}}_{W}$ reduces to $\mathcal{S}$-LH. But, before describing W, we must analyze $\mathcal{S}$.

Denote the assumed 2-qubit symmetric term that is not 1-local in $\mathcal{S}$ by H. As in the previous proof, without changing the complexity of $\{H\}$-PRODLH we may conjugate and scale as necessary so that

$$H^{ab} = \alpha^- X^a X^b + \beta^- Y^a Y^b + \gamma^- Z^a Z^b + \sum_{j=1}^{3} v_j(\sigma_j^a I^b + I^a \sigma_j^b),$$

where all coefficients are real and at least one of $\alpha^-, \beta^-, \gamma^-$ is nonzero since H is nonzero. The superscripts in the above equations are to differentiate the coefficients of H from the entries of W, which must be non-negative.

We again use a symmetrization gadget to remove the 1-local terms, but now require negative weights. Given four qubits a, b, c, d, define $B = H^{ab} + H^{cd} - H^{bd} - H^{ac}$. This is a rectangle with two positive edges and two negative edges. Then

$$B = \alpha(X^a - X^d)(X^b - X^c) + \beta(Y^a - Y^d)(Y^b - Y^c) + \gamma(Z^a - Z^d)(Z^b - Z^c).$$

Now we see how B interacts with product states on four qubits. For $e = a, b, c, d$, we again define $\rho^e = \frac{1}{2}(I + r^e \cdot v^e)$ with $v^e = (X^e, Y^e, Z^e)$ and $r^e = (x^e, y^e, z^e)$. Then, writing any product state on a, b, c, d as $\rho^a \rho^b \rho^c \rho^d$, the expectation value on B is $\mathrm{tr}\left[B \rho^a \rho^b \rho^c \rho^d\right]$, which equals

$$\alpha^-(x^a - x^d)(x^b - x^c) + \beta^-(y^a - y^d)(y^b - y^c) + \gamma^-(z^a - z^d)(z^b - z^c)$$

which is in turn equal to

$$\left(r^b - r^c\right)^\top W' \left(r^a - r^d\right) \quad \text{for} \quad W' = \mathrm{diag}\left(\alpha^-, \beta^-, \gamma^-\right).$$

If we fix the state of qubits a and d and minimize the expectation on B, the minimum is achieved when $r^b = -W'(r^a - r^d)/\|W'(r^a - r^d)\|$ and $r^c = -r^b$. This minimum expectation is $-2\|W'(r^a - r^d)\|$. Observe this expectation value is equivalent to $-2\|W(r^a - r^d)\|$ for $W = \mathrm{diag}(\alpha, \beta, \gamma)$ where $\alpha = |\alpha^-|, \beta = |\beta^-|, \gamma = |\gamma^-|$.

Now we are prepared to set up our reduction. For $W = \mathrm{diag}(\alpha, \beta, \gamma)$, which is non-negative and nonzero, consider an arbitrary instance of $\mathrm{MC}^{\mathrm{L}}_W$. For a given graph $G = (V, E)$, the objective function is again $\mathrm{MC}^{\mathrm{L}}_W(G)$, and given a parameter C, we must decide whether $\mathrm{MC}^{\mathrm{L}}_W(G)$ is at least C or at most $C - \varepsilon$.

Associate a "vertex qubit" with each vertex and construct a copy of the gadget B for each edge ij, such that it acts on i, j and two ancilla qubits, and denote it B^{ij}. The vertex qubits may be shared among several gadgets, while the ancilla qubits are part of only one gadget. In particular, we choose a and d in each gadget to be the vertex qubits.

Substituting our gadget for every edge constructs a Hamiltonian H_{final} such that the minimum expectation of any product state is equal to

$$2 \min_{r^i \in S^2} \sum_{ij \in E} -\|Wr^i - Wr^j\| = -2 \max_{r^i \in S^2} \sum_{ij \in E} \|Wr^i - Wr^j\|.$$

Simply multiplying H_{final} by $\frac{1}{2}$ makes this equal to $-\mathrm{MC}^{\mathrm{L}}_W(G)$.

We conclude that given $\mathcal{S}$ contains a 2-qubit symmetric term H that is not 1-local, there exists some non-negative $W = \mathrm{diag}(\alpha, \beta, \gamma)$ with at least one nonzero entry such that $\mathrm{MC}^{\mathrm{L}}_W(G)$ reduces to $\textsc{prodlh}(H_{\text{final}})$. Since H_{final} was constructed using only the symmetric term $H \in \mathcal{S}$, this is also a reduction to an instance of $\mathcal{S}$-$\textsc{prodlh}$, as desired. ◀

4 The Stretched Linear Max-Cut Problem

We study a generalization of the classical Max-Cut problem which arises naturally from our study of product states and which is likely of independent interest. Both Max-Cut and its generalization

$$\mathrm{MC}_k(G) = \frac{1}{2} \max_{\hat{\imath} \in S^{k-1}} \sum_{ij \in E} 1 - \hat{\imath} \cdot \hat{\jmath} = \frac{1}{4} \max_{\hat{\imath} \in S^{k-1}} \sum_{ij \in E} \|\hat{\imath} - \hat{\jmath}\|^2$$

were introduced in Section 1. As the above equation emphasizes, maximizing the distance between vectors is equivalent to maximizing the angle, optimally being anti-parallel.

Our new problem is defined with two significant changes. First, the sum is over distances rather than squared distances. Second, the distances are allowed to incorporate a linear stretch.

▶ **Definition 20** (W-linear-Max-Cut ($\mathrm{MC}^{\mathrm{L}}_W$)). *For a fixed $d \times d$ diagonal matrix W, given an n-vertex graph $G = (V, E)$ and thresholds $b > a \geq 0$ with $b - a \geq 1/\mathrm{poly}(n)$, decide whether*

$$\mathrm{MC}^{\mathrm{L}}_W(G) = \frac{1}{2} \max_{\hat{\imath} \in S^{d+1}} \sum_{ij \in E} \|W\hat{\imath} - W\hat{\jmath}\| = \frac{1}{2} \max_{\hat{\imath} \in S^{d+1}} \sum_{ij \in E} \sqrt{\|W\hat{\imath}\| + \|W\hat{\jmath}\| - 2(W\hat{\imath})^\top (W\hat{\jmath})}$$

is at least b or at most a.

A comparison of the geometric interpretations of MC_k and $\mathrm{MC}^{\mathrm{L}}_W$ was given in Section 1. A further interpretation comes from treating the edges of the graph as springs or rubber bands. As explored in [29], the potential energy of a spring is quadratic in its length, so the MC_k value represents the total potential energy of the system given a particular embedding. On

the other hand, the force or tension of each spring is linear in its length. So, $\mathrm{MC}^{\mathrm{L}}_W$ gives the total force, tension, or pressure such an arrangement of springs would apply to the surface of the sphere (or ellipsoid, more generally).

In both problems, the objective is a linear sum, of either the distances or the inner products. Both problems generalize the traditional Max-Cut problem, since when restricted to ± 1 labels, distances are directly proportional to squared distances. Previous work was likely motivated to focus on squared distances because approximation algorithms like SDPs naturally apply to inner products but not to square roots of inner products.

Our main theorem concerning W-linear-Max-Cut is the below.

▶ **Theorem 4.** *For any fixed non-negative $W = \mathrm{diag}(\alpha, \beta, \gamma)$ with at least one of α, β, γ nonzero, W-linear-Max-Cut is NP-complete.*

Our hardness proofs do not require any edge weights (unlike our Hamiltonian constructions in the previous section).

Containment in NP is immediate, and we break the proof of NP-hardness into three cases based on the entries of W. The three cases depend on how many entries of W are equal, requiring different approaches for dealing with degenerate solutions. We assume throughout that $1 = \alpha \geq \beta \geq \gamma$; as we show in the final proof of Theorem 4, this suffices by scaling and symmetry. Lemma 21 considers the case when all three entries are equal. Lemma 22 considers the case when the largest entry is unique. Lemma 23 finally considers the case when the two largest entries are equal and distinct from the third, combining techniques from the previous two proofs. Note that these cases are not entirely disjoint.

When $W = \mathrm{diag}(1, 1, 1)$, we prove hardness by reducing from the NP-complete 3-Coloring problem. We replace every edge in the graph with a 4-clique, or tetrahedron. To deal with the symmetry created by equally weighted axes, all of the gadgets are connected to a new *sink* vertex, removing a degree of freedom. We then argue that there is an assignment to the new graph that simultaneously (nearly) maximizes all of these cliques iff the original graph is 3-colorable.

▶ **Lemma 21.** *For $W = \mathrm{diag}(1, 1, 1)$, W-linear-Max-Cut is NP-hard.*

Proof. We will reduce from 3-Coloring. Consider an arbitrary graph $G = (V, E)$ on n vertices and m edges. We construct $H = (V', E')$ as follows: Start with G. For each edge $ij \in E$, add vertices k_{ij}, t_{ij} and connect i, j, k_{ij}, t_{ij} to form a 4-clique. Then add a *sink* vertex t and an edge tt_{ij} for each t_{ij}. H therefore consists of m edge-disjoint 4-cliques, each containing one edge from G, and m additional edges from vertices t_{ij} to t.

We claim that if G is 3-colorable, then $\mathrm{MC}^{\mathrm{L}}_W(H) \geq m\mathrm{MC}^{\mathrm{L}}_W(K_4) + m$. Conversely, we claim that if $\mathrm{MC}^{\mathrm{L}}_W(H) \geq m\mathrm{MC}^{\mathrm{L}}_W(K_4) + m - \varepsilon$, for an $\varepsilon = \Omega\big(1/m^2\big)$ we will choose later, then G is 3-colorable.

First, suppose G is 3-colorable. We will show how to derive a vector assignment to H attaining $m\mathrm{MC}^{\mathrm{L}}_W(K_4) + m$ from a 3-coloring of G. Define

$$\Delta_W(u, v, w, r) = \frac{1}{2} \sum_{ij \in \{uv, uw, ur, vw, vr, wr\}} \|Wi - Wj\|$$

Let u, v, w, r be vectors corresponding to a regular tetrahedron inscribed in the unit sphere, known to achieve the maximum perimeter of any inscribed tetrahedron at $4\sqrt{6}$ [30]. We 3-color G (and therefore the vertices of H other than $(k_{ij})_{ij \in E}$, $(t_{ij})_{ij \in E}$, and t) with u, v, w, assigning each vertex the vector matching its color. Then for each $ij \in E$, we assign k_{ij}

the vector in $\{u, v, w\}$ not assigned to i or j, and r to t_{ij}. Finally, we assign $-r$ to t. By construction, this assignment will yield a value of $\mathrm{MC}^{\mathrm{L}}_W(K_4)$ on each 4-clique gadget, and the edges tt_{ij} will each contribute exactly 1. Thus $\mathrm{MC}^{\mathrm{L}}_W(H) \geq m\mathrm{MC}^{\mathrm{L}}_W(K_4) + m$, as desired.

Now we will show the converse. Suppose there exists an assignment $(\hat{k})_{k\in V'}$ of vectors achieving greater than $m\mathrm{MC}^{\mathrm{L}}_W(K_4)+m-\varepsilon$ on H. We construct a 4-coloring of each connected component of $H \setminus \{t\}$ as follows: Choose an edge ij in the corresponding component of G (note that there is a 1-to-1 correspondence between components of G and $H \setminus \{t\}$). Let $\hat{\imath}, \hat{\jmath}, \hat{k}_{ij}, \hat{t}_{ij}$ be the vectors assigned to the vertices of the corresponding clique in H. Our coloring will use these four vectors as colors, which we denote as the set $\mathcal{C}$. We assign each vertex v the color corresponding to the element of $\mathcal{C}$ that is closest to the vector assigned to v in the $\mathrm{MC}^{\mathrm{L}}_W$ assignment. We will show that this is a proper coloring, and that it assigns the same color to every t_{ij}, and therefore gives a 3-coloring of G.

To show that this is a proper coloring, we need to show that every pair of adjacent vertices in $H \setminus \{t\}$ are assigned different colors, i.e. that the closest elements of $\mathcal{C}$ to the vectors assigned to them in the $\mathrm{MC}^{\mathrm{L}}_W$ assignment are distinct. As every edge in $H \setminus \{t\}$ is contained in some 4-clique corresponding to some edge xy of G, it will suffice to show the following: for every edge xy in the component of G containing ij, if d is the smallest number of edges in a path in G starting with ij and ending with xy, each of $\hat{x}$, $\hat{y}$, $\hat{k}_{xy}$ and $\hat{t}_{xy}$ is within $\mathrm{O}(d\sqrt{\varepsilon})$ of a *different* element of $\mathcal{C}$.

First we note that, as H consists of m edge-disjoint 4-cliques and m other edges, and the maximum any assignment can earn on an edge is 1, the lower bound on the total amount the assignment earns implies that every 4-clique earns at least $\mathrm{MC}^{\mathrm{L}}_W(K_4) - \varepsilon$ and the other edges $(t_{xy}t)_{xy\in E}$ earn at least $1-\varepsilon$ each. So by trigonometry we immediately have that every $\hat{t}_{xy}$ is within $\mathrm{O}(\sqrt{\varepsilon})$ of $-\hat{t}$, and therefore within $\mathrm{O}(\sqrt{\varepsilon})$ of each other, and $\hat{t}_{ij}$ in particular.

For the other vertices, we proceed by induction on d. We have the desired result trivially for $d = 1$, as in this case $ij = xy$. Now suppose it holds for d. Let xy be the end of a $(d+1)$-edge path starting with ij. Without loss of generality let y be the vertex of xy earlier in the path, and let z be the immediately prior vertex in the path, so by the inductive hypothesis $\hat{y}$, $\hat{z}$, $\hat{k}_{yz}$, and $\hat{t}_{yz}$ are within $\mathrm{O}(d\sqrt{\varepsilon})$ of different elements of $\mathcal{C}$. Furthermore, as both $\{\hat{x}, \hat{y}, \hat{k}_{xy}, \hat{t}_{xy}\}$ and $\{\hat{y}, \hat{z}, \hat{k}_{yz}, \hat{t}_{yz}\}$ are vertices of tetrahedra with perimeters at least $4\sqrt{6} - \varepsilon$, by Lemma 29 the edge lengths of these tetrahedra are all in the interval $[\frac{4\sqrt{6}}{6} - \mathrm{O}(\sqrt{\varepsilon}), \frac{4\sqrt{6}}{6} + \mathrm{O}(\sqrt{\varepsilon})]$. So as we have already shown that $\hat{t}_{yz}$, $\hat{t}_{xy}$, and $\hat{t}_{ij}$ are all within $\mathrm{O}(\sqrt{\varepsilon})$ of each other, then the criteria of Lemma 30 are satisfied with $ABCD = \hat{y}\hat{t}_{xy}\hat{x}\hat{k}_{xy}$ and $AEFG = \hat{y}\hat{t}_{yz}\hat{z}\hat{k}_{yz}$ and so $\hat{x}$ and $\hat{k}_{xy}$ are each within $\mathrm{O}(\sqrt{\varepsilon})$ of (different) elements of $\{\hat{z}, \hat{k}_{yz}\}$. So then the result follows by the triangle inequality.

We now have that every vertex in H is assigned a vector within $\mathrm{O}(m\sqrt{\varepsilon})$ of an element of $\mathcal{C}$, and so taking ε to be a sufficiently small constant times $1/m^2$, we can take these distances to be at most 0.1. Moreover, for any two adjacent vertices, the choice of color will be different, and so as by Lemma 29 the distances between these four vectors are at least $4/\sqrt{6} - \mathrm{O}(\varepsilon)$, this implies any two adjacent vertices are assigned different colors, and so we have a proper 4-coloring of $H \setminus \{t\}$. Finally, note that, as every $\hat{t}_{xy}$ is within $\mathrm{O}(\varepsilon)$ of every other one, this implies all the t_{xy} were assigned the same color, and so no vertex in G uses this color. So this gives us a proper 3-coloring of G. ◀

Next, in Lemma 22 we consider the case $W = \mathrm{diag}(1, \beta, \gamma)$ and $1 > \beta, \gamma \geq 0$, in which the maximum weight is unique. The approach in the proof of Lemma 21, reducing from 3-Coloring, can in fact be modified to cover this case, but analyzing triangles inscribed in ellipses instead of circles is more technical. Instead, we take a different approach and in the case of a unique maximum (whether β equals γ or not) give a reduction from Max-Cut instead of 3-Coloring.

We insert ancilla vertices so that every vertex in the original graph is the center of a large star. These star gadgets amplify any deviation from the highest-weight axis such that any near-optimal solution must approximate a standard 1-dimensional labeling, as in Max-Cut.

▶ **Lemma 22.** *For any* $W = \mathrm{diag}(1, \beta, \gamma)$ *with* $1 > \beta \geq \gamma \geq 0$, W-linear-Max-Cut *is* NP*-hard.*

Proof. We reduce from the standard NP-complete Max-Cut problem. For any graph $G = (V, E)$ with n vertices and m edges, we construct a graph $H = (V', E')$ with $V \subseteq V'$ and $E \subseteq E'$ by, for each $v \in V$, adding $K = m^3 n$ ancilla vertices v_i, and then adding an edge from each of these vertices to v so that v is the center of a K-star. Now $|V'| = n(1+K)$ and $|E'| = m + Kn$.

We claim that, for any $C > 1$ and for large enough n, $\mathrm{MC}(G) \geq C$ implies $\mathrm{MC}^{\mathrm{L}}_W(H) \geq C + Kn$ and $\mathrm{MC}^{\mathrm{L}}_W(H) > C + Kn - 1/2$ implies $\mathrm{MC}(G) \geq C$.

First, suppose there is a cut of G with value at least C. We construct a corresponding assignment of vectors to vertices in V'. First assign the vector $(1, 0, 0)$ to all vertices in V which are labeled $+1$ in G and $(-1, 0, 0)$ to those with labels -1. Then, for every vertex in $v \in V$, which by construction is at the center of a star of ancilla qubits in H, assign the vector opposite the one assigned to v to each of the ancilla vertices. This assignment of vectors gives an objective value of at least C on the edges from the original graph and Kn on the edges of the star gadgets, and so the $\mathrm{MC}^{\mathrm{L}}_W$ value of this assignment is $C + Kn$.

Now suppose there exists an assignment of vectors achieving $\mathrm{MC}^{\mathrm{L}}_W$ value greater than $C + Kn - 1$ on H. We will show that the cut given by $\mathrm{sgn}(\hat{v}_1)$ for each $v \in V$ (i.e. projecting $\hat{v}$ to the x-axis and checking whether it is ≥ 0 or < 0) has value at least C.

First, for each $v \in V$, let $\widehat{\mathrm{sgn}}(\hat{v}) = (\mathrm{sgn}(\hat{v}_1), 0, 0)$. We will show that this is close to $\widehat{v}$. Because the original graph can contribute at most m to the $\mathrm{MC}^{\mathrm{L}}_W$ objective, and each star gadget can contribute at most K, each star gadget must contribute at least $K + (C - 1/2 - m) \geq K - 1/2 - m > K\big(1 - \frac{2m}{K}\big)$. By Lemma 26, for the K-star to achieve at least $K\big(1 - \frac{2m}{K}\big)$, the vector $\hat{v}$ assigned to v must satisfy

$$\|W\hat{v} - \widehat{\mathrm{sgn}}(v)\| \leq \delta \quad \text{for} \quad \delta = 2\sqrt{\frac{2m}{K}}\sqrt{\frac{1+\beta^2}{1-\beta^2}}.$$

We will use this fact to show that the $\mathrm{MC}^{\mathrm{L}}_W$ value earned by the vector assignment on the original graph G is close to the value of the cut we defined. We have

$$\left|\sum_{ij\in E} \|W\hat{\imath} - W\hat{\jmath}\| - \sum_{ij\in E} \|\widehat{\mathrm{sgn}}(\hat{\imath}) - \widehat{\mathrm{sgn}}(\hat{\jmath})\|\right| = \left|\sum_{ij\in E} (\|W\hat{\imath} - W\hat{\jmath}\| - \|\widehat{\mathrm{sgn}}(\hat{\imath}) - \widehat{\mathrm{sgn}}(\hat{\jmath})\|)\right|$$
$$\leq \left|\sum_{ij\in E} (\|W\hat{\imath} - \widehat{\mathrm{sgn}}(\hat{\imath})\| + \|W\hat{\jmath} - \widehat{\mathrm{sgn}}(\hat{\jmath})\|)\right| \leq 2m\delta = 4\sqrt{\frac{2}{n}}\sqrt{\frac{1+\beta^2}{1-\beta^2}} = \mathrm{O}\big(1/\sqrt{n}\big)$$

which is $< 1/2$ for large enough n.

Using for a second time the fact that the edges of the star gadgets can contribute at most Kn to the $\mathrm{MC}^{\mathrm{L}}_W(H)$, the vector assignment must achieve at least $C - 1/2$ on G, and so

$$C - 1/2 \leq \frac{1}{2}\sum_{ij\in E} \|W\hat{\imath} - W\hat{\jmath}\| \leq \frac{1}{2}\sum_{ij\in E} |\widehat{\mathrm{sgn}}(\hat{\imath}) - \widehat{\mathrm{sgn}}(\hat{\jmath})| + \mathrm{O}\big(1/\sqrt{n}\big)$$

implying that the value of our cut is strictly greater than $C - 1$ for sufficiently large n. Therefore, as it is integer-valued it is at least C, concluding the proof. ◀

Finally, we give Lemma 23, in which $W = \mathrm{diag}(1, 1, \gamma)$ and $1 > \gamma$. Our proof combines the techniques in the previous two proofs. Similar to the proof of Lemma 21, we show hardness by reducing from 3-Coloring. Our construction begins by replacing every edge in the graph with a 3-clique. Then, as in the proof of Lemma 22, we insert large star gadgets on every vertex, forcing solutions away from the low-weight z-axis. With solutions restricted to two dimensions, we are able to argue there is an assignment to the new graph that simultaneously (nearly) maximizes all of the cliques iff the original graph was 3-colorable.

▶ **Lemma 23.** *For any $W = \mathrm{diag}(1, 1, \gamma)$ with $1 > \gamma \geq 0$, W-linear-Max-Cut is* NP*-hard.*

Proof. Consider an arbitrary instance of 3-Coloring on a graph $G = (V, E)$ with n vertices and m edges. Construct a new graph $H' = (V', E')$ by taking G and, for each edge $ij \in E$, adding a vertex k_{ij} and edges ik_{ij} and jk_{ij}, so that each edge in G corresponds to a 3-clique (K_3) in H'. Note that the m cliques constructed this way are edge-disjoint. Next, for each vertex v in H', add $K = m^6$ ancilla vertices v_i, each connected to v so that v is the center of a K-star. Call the final graph $H'' = (V'', E'')$, for which we have $(K + 1)(n + m)$ vertices and $3m + K(n + m)$ edges.

We claim that if G is 3-colorable, then $\mathrm{MC}^{\mathrm{L}}_W(H'') \geq K(n + m) + 3\sqrt{3}m$. Conversely, we claim that if $\mathrm{MC}^{\mathrm{L}}_W(H'') > K(n + m) + 3\sqrt{3}m - \varepsilon$, for an $\varepsilon = \Omega(1/m^2)$ we will choose later, then G is 3-colorable. As testing 3-colorability is NP-hard, this will prove the theorem.

First, suppose G is 3-colorable. Let C be any set of three vectors in the xy-plane achieving the maximum value of $\mathrm{MC}^{\mathrm{L}}_W(K_3) = 3\sqrt{3}$. Given any 3-coloring of G, we assign one of these vectors to each color, and thus assign vectors from C to G with no two adjacent vertices having the same vector. We extend this assignment to H' by, for each of our constructed 3-cliques i, j, k_{ij}, assigning the vector in C that was not assigned to either i or j to k_{ij}. Now each of these cliques contributes $3\sqrt{3}$ to the objective, and as there are m of them and they are edge-disjoint, this contributes $3\sqrt{3}m$ to the objective. To extend to H'' and its star gadgets, every edge of a star centered on vertex v can trivially contribute $\|v\|$ to the objective value. Because all vectors are in the xy-plane, and so the unit circle, this means they all contribute 1. So in total, there exists an assignment with value $3\sqrt{3}m + K(n + m)$.

Now suppose there exists a vector assignment achieving greater than $K(n+m)+3\sqrt{3}m-\varepsilon$ on H''. For a vector $\hat{v} = (\hat{v}_x, \hat{v}_y, \hat{v}_z)$, let $\widehat{\mathrm{sgn}}(\hat{v})$ denote the vector $(\hat{v}_x, \hat{v}_y, 0)/\|(\hat{v}_x, \hat{v}_y, 0)\|$. We assign colors as follows. Choose any of the K_3 gadgets i, j, k_{ij} in H'' corresponding to an edge ij in the original graph. Let $\hat{\imath}, \hat{\jmath}, \hat{k}_{ij}$ be the vectors assigned to the vertices, respectively. Assign three colors to the vertices. Then, choose any K_3 gadget $i, \ell, k_{i\ell}$ adjacent to the first. For each vertex in the second clique, consider its assigned vector and round it via $\widehat{\mathrm{sgn}}$, determine which of the rounded vectors $\widehat{\mathrm{sgn}}(\hat{\imath}), \widehat{\mathrm{sgn}}(\hat{\jmath}), \widehat{\mathrm{sgn}}(\hat{k}_{ij})$ it is closest to, and assign the same color. We continue coloring adjacent cliques in this way, comparing rounded vectors to the original set $\widehat{\mathrm{sgn}}(\hat{\imath}), \widehat{\mathrm{sgn}}(\hat{\jmath}), \widehat{\mathrm{sgn}}(\hat{k}_{ij})$, until the coloring is propagated to the entire graph. This colors all of the vertices in H', which are the centers of star gadgets in H''. We will show no adjacent vertices in H' were assigned the same color. Since G is a subgraph of H', this also implies a proper coloring of G, as desired.

We first show that just as the assigned vectors must achieve a large objective value on the clique gadgets, the star gadgets force the rounded vectors to achieve a similar score. Because the m clique gadgets can each contribute at most $3\sqrt{3}$ to the objective value, the star gadgets in H'' must contribute at least $K(n + m) - \varepsilon$. Similarly, because the edges of each star can contribute at most K, each star must achieve at least $K - \varepsilon$. For any vertex in s in H', consider the star gadget centered on it in H''. As shown in Lemma 26, for the star to achieve $K(1 - \varepsilon/K)$, the vector $\hat{s}$ assigned to s must satisfy

$$\|W\hat{s} - \widehat{\operatorname{sgn}}(\hat{s})\| \leq \delta \quad \text{for} \quad \delta = 2\sqrt{\frac{\varepsilon}{K}}\sqrt{\frac{1+\gamma^2}{1-\gamma^2}}.$$

On the other hand, because the star gadgets can contribute at most $K(n+m)$ to the objective value, the vector assignment must achieve at least $3\sqrt{3}m - \varepsilon$ on the remaining edges, the ones in H'. Given the vectors are close, we have that

$$\left|\sum_{ij\in E'} \|W\hat{\imath} - W\hat{\jmath}\| - \sum_{ij\in E'} \|\widehat{\operatorname{sgn}}(\hat{\imath}) - \widehat{\operatorname{sgn}}(\hat{\jmath})\|\right| = \left|\sum_{ij\in E'} \|W\hat{\imath} - W\hat{\jmath}\| - \|\widehat{\operatorname{sgn}}(\hat{\imath}) - \widehat{\operatorname{sgn}}(\hat{\jmath})\|\right|$$

$$\leq \left|\sum_{ij\in E'} \|W\hat{\imath} - \widehat{\operatorname{sgn}}(\hat{\imath})\| + \|W\hat{\jmath} - \widehat{\operatorname{sgn}}(\hat{\jmath})\|\right| \leq 2|E'|\delta = 12\sqrt{\varepsilon}m^{-2}.$$

Therefore, for $\mu = \varepsilon + 12\sqrt{\varepsilon}m^{-2}$, the set of rounded vectors must achieve at least $3\sqrt{3}m - \mu$ on the rest of H'. Because each of the m clique gadgets can contribute at most $3\sqrt{3}$, the rounded vectors must achieve at least $3\sqrt{3} - \mu$ on each individual clique.

The rounded vectors exist in the xy-plane, which means they are inscribed in the unit circle. Maximizing the sum of the edge lengths on a K_3 gadget is equivalent to maximizing the perimeter of a triangle. For a triangle in the unit circle to have nearly maximal perimeter, it must be nearly regular; as shown in Lemma 27, if the perimeter is at least $3\sqrt{3} - \mu$, then each edge length must be in the interval $\sqrt{3} \pm 4\sqrt{\mu}$.

Now, consider any two adjacent vertices u, v in H', which are at the center of star gadgets in H''. We must show they were assigned different colors. The two vertices exist in some K_3 gadget in H', and the coloring procedure, starting from i, j, k_{ij}, must have reached them in at most m rounds. At each step in the procedure, there is a colored clique and a successive adjacent clique which share a clique. Two nearly regular triangles which share a vertex must nearly share their other vertices; as shown in Lemma 28, the distance between the first clique's rounded vectors and the second clique's is at most $\mathrm{O}\big(\sqrt{\mu}\big)$.

After m rounds of the coloring procedure, we conclude u and v must be assigned colors such that $\widehat{\operatorname{sgn}}(\hat{u}), \widehat{\operatorname{sgn}}(\hat{v})$ are each at distance at most $m \times \mathrm{O}\big(\sqrt{\mu}\big) = \mathrm{O}\big(\varepsilon^{1/4}\big)$ away from the same-colored rounded vectors in the initial clique. Both the initial clique and the clique containing u and v must have vectors separated by at least $\sqrt{3} - \varepsilon$. For sufficiently small ε ($\varepsilon = \Theta\big(1/m^2\big)$ with a sufficiently small constant suffices), $\widehat{\operatorname{sgn}}(\hat{u}), \widehat{\operatorname{sgn}}(\hat{v})$ cannot also be within $\mathrm{O}(\sqrt{\varepsilon})$ of the same initial vector, so we conclude they must be associated with different colors, and this is a proper coloring. ◀

We now conclude with a proof of the main theorem of this section, that W-LINEAR-MAX-CUT is NP-complete for any fixed diagonal 3×3 non-negative nonzero W.

Proof of Theorem 4. The containment of W-LINEAR-MAX-CUT for any diagonal W is straightforward. With W a constant, given a claimed vector assignment, the value $\mathrm{MC}^{\mathrm{L}}_W(G)$ can be verified in time linear in the number of edges.

To show hardness, we make two simplifications. First, because $\mathrm{MC}^{\mathrm{L}}_{cW} = c\mathrm{MC}^{\mathrm{L}}_W$ for a constant c, we can easily reduce to an instance in which we assume the largest entry of W equals 1. Second, although rearranging the entries of diagonal W requires changing any vector assignment, it does not change the objective value. So $\mathrm{MC}^{\mathrm{L}}_{\mathrm{diag}(\alpha,\beta,\gamma)} = \mathrm{MC}^{\mathrm{L}}_{\mathrm{diag}(\beta,\alpha,\gamma)}$ and any other rearrangement of W, and we can assume the entries are ordered $1 \geq \alpha \geq \beta \geq \gamma \geq 0$. With this, the theorem follows by Lemmas 21–23. ◀

5 NP-Hardness of Unweighted Quantum Max-Cut

Our lower bounds elsewhere in this paper are for local Hamiltonian problems in which terms can be given positive *or* negative weight. They apply in the model where all weights are restricted to being of constant weight but require some terms with negative weight to work. In this section we show that this restriction can be removed for one of the best-studied local Hamiltonian problems: the Quantum Max-Cut (or QMC) problem.

The Quantum Max-Cut problem can be defined as $\mathcal{S}$-LH with $\mathcal{F} = \{XX + YY + ZZ\}$. We will write h for $XX + YY + ZZ$.[6]

▶ **Definition 24** (Quantum Max-Cut (QMC)). *Given a Hamiltonian $H = \sum_{ij}^{n} w_{ij} H_{ij}$ acting on n qubits where each $H_{ij} = I - h_{ij}$, all w_{ij} are real, polynomially-bounded, and specified by at most* $\mathrm{poly}(n)$ *bits, and two real parameters b, a such that $b - a \geq 1/\mathrm{poly}(n)$, decide whether $\lambda_{\max}(H)$ is at least b (YES) or at most a (NO).*

Note that we have written QMC as a *maximum* eigenvalue problem (with a flip and shift of the local terms) rather than in terms of the minimum eigenvalue as for k-LH in Definition 7. This is to follow the norm in previous work; note that as both the terms *and* the objective function are flipped, an instance of the problem defined this way will be equivalent to an instance of the corresponding k-LH problem with the same weights.

When minimizing (maximizing) the eigenvalue and the weights are restricted to be non-negative (non-positive), it is referred to as the anti-ferromagnetic Heisenberg model. Flipping the restrictions, e.g. minimizing with non-positive weights, is referred to as the ferromagnetic Heisenberg model. The latter case is trivial when viewed as an optimization problem (as $|00\rangle$ earns 0 on the local term, and so the problem is optimized by assigning $|0\rangle$ to every qubit), so we will be interested in hardness results for the former.

It is straightforward to verify that for two qubits a, b with *pure* states ρ^a, ρ^b,

$$\mathrm{tr}\left(\rho^a \otimes \rho^b \; h\right) = r^a \cdot r^b = 1 - \frac{1}{2}\left\|r^a - r^b\right\|^2,$$

where r^a, r^b are the corresponding Bloch vectors. This shows that deciding QMC restricted to product states, which we denote prodQMC, is equivalent to the standard Vector Max-Cut problem in three dimensions:

▶ **Definition 25** ($\text{Max-Cut}_k(\text{MC}_k)$). *Given an n-vertex graph $G = (V, E)$ and thresholds $b > a \geq 0$ such that $b - a \geq 1/\mathrm{poly}(n)$, decide whether*

$$\text{MC}_k = \frac{1}{2} \max_{\hat{\imath} \in S^{k-1}} \sum_{ij \in E} 1 - \hat{\imath} \cdot \hat{\jmath} = \frac{1}{4} \max_{\hat{\imath} \in S^{k-1}} \sum_{ij \in E} \left\|\hat{\imath} - \hat{\jmath}\right\|^2$$

is at least b or less than a.

Note that this is different from the W-linear-Max-Cut we studied in Section 4 as it considers *squared* distances. Furthermore, while Max-Cut is a classic NP-complete problem, MC_k is not expected to be hard for all values of k, and in particular is tractable when $k = n = |V|$.

Our main result classifying $\mathcal{S}$-prodLH immediately implies $\{h\}$-prodLH, and therefore also prodQMC and MC_3, is NP-complete. However, the proof of Lemma 15 utilizes Hamiltonian gadgets involving negative weights. This leaves open whether prodQMC and

[6] Other work frequently includes a multiplicative factor, e.g. 1/2, in the definition of h and/or of QMC.

MC_3 remain NP-hard on unweighted graphs. We now prove that MC_3 is NP-hard even when restricted to positive unit weights. This is the first published proof of this fact, although we note that a sketch of a different proof was previously known for this specific problem [40].

Our approach is similar to the proof of Lemma 21, which demonstrated hardness of $\mathrm{MC}_W^{\mathrm{L}}$ for $W = I$ by replacing every edge with a 4-clique (with one vertex connected to a source vertex), and showing that the resulting graph could simultaneously optimize all of these 4-cliques (by assigning vectors corresponding to a regular tetrahedron) if and only if the original graph was 3-colorable.

However, the change in objective function from distances to *squared* distances causes a problem: while the regular tetrahedron is the unique optimal solution for $\mathrm{MC}_W^{\mathrm{L}}(K_4)$, in the case of $\mathrm{MC}_3(K_4)$, setting $v_1 = v_2 = (1, 0, 0), v_3 = v_4 = (-1, 0, 0)$ would also be optimal. So we replace each edge of the 4-cliques with triangles, which penalize the degenerate solution and forces the vectors assigned to the vertices of the 4-cliques toward regular tetrahedra.

▶ **Theorem 5.** MC_3 *is* NP*-complete.*

Proof of Theorem 5. Clearly MC_3 is in NP. To show hardness, we reduce 3-Coloring to MC_3. Given an instance $G = (V, E)$ on n vertices and m edges, we first replace every edge with a copy of K_4. That is, for each edge $ij \in E$, we add vertices q_{ij}, t_{ij} and add edges to form a 4-clique. Call the new graph $H' = (V', E')$. Then in H', we replace every edge with a copy of K_3. That is, for all $ij \in E'$ we add a vertex r_{ij} and edges ir_{ij} and jr_{ij}. Finally, we add a *sink* vertex t and for every edge $ij \in E$, in the original graph, add edge tt_{ij}. Call the resulting graph $H'' = (V'', E'')$. For future reference, let R denote a copy of K_4 with each edge replaced by a copy of K_3, which we may call a "tetrahedron with adjoined triangles".

We claim that if G has a proper 3-coloring then $\mathrm{MC}_3(H'') \geq m\,\mathrm{MC}_3(R) + m$. Conversely, we claim that if $\mathrm{MC}_3(H'') \geq m\,\mathrm{MC}_3(R) + m - \varepsilon$, for an $\varepsilon = \Omega\big(1/m^2\big)$ we will choose later, then there is a 3-coloring of G. Later, we will show $\mathrm{MC}_3(R) = 10 + 2\sqrt{3}$.

First, suppose G is 3-colorable. Let S consist of the following three unit vectors in $\mathbb{R}^3$:

$$\left(\sqrt{8/9}, 0, -1/3\right), \left(-\sqrt{2/9}, \sqrt{2/3}, -1/3\right), \left(-\sqrt{2/9}, -\sqrt{2/3}, -1/3\right).$$

Along with $(0, 0, 1)$, these are the four vertices of a regular tetrahedron inscribed in the unit sphere. Assign each one of the vectors of S to each color, such that every vertex in G is assigned a vector and no adjacent vertices have the same vector. We copy those vectors to the vertices of H', and for each vertex q_{ij} we assign the vector in S not assigned to i or j. We assign $(0, 0, 1)$ to each vertex t_{ij}. We copy these vectors to the vertices of H''. In H'' we assign $(0, 0, -1)$ to t. Finally, for each edge $ij \in E'$ and 3-clique i, j, r_{ij}, the only uncolored vertex is r_{ij}, and we assign the unique unit vector that is antiparallel to the sum of the vectors assigned to i and j.

Now we calculate the objective value that this assignment achieves. For vertex i, let $\hat{\imath}$ denote the assigned vector. For any edge ij in G, the vectors assigned to the associated K_4 gadget correspond to vertices of a regular tetrahedron, and it can be directly calculated that for any of the six edges ab, we have $\hat{a} \cdot \hat{b} = -1/3$. For edges tt_{ij}, we have $t \cdot t_{ij} = -1$. For edges ir_{ij}, given $\hat{\imath} \cdot \hat{\jmath} = -1/3$ and $\hat{r}_{ij} \propto -(\hat{\imath} + \hat{\jmath})$, the inner product is $\hat{\imath} \cdot \hat{r}_{ij} = \hat{\jmath} \cdot \hat{r}_{ij} = -1/\sqrt{3}$. In total, for each edge ij in G, the graph H'' has a gadget with six edges in a K_4 gadget, twelves edges in K_3 gadgets, and one edge incident on t. Plugging the inner products we calculated into the definition of MC_k gives a total objective value of $m(10 + 2\sqrt{3}) + m$.

Second, suppose there is a set of unit vectors $\hat{\imath}$ for $i \in V''$ such that $\mathrm{MC}_3(H'') \geq m\,\mathrm{MC}_3(R) + m - \varepsilon$. We color the vertices of H', comprised of the K_4 gadgets, as follows. Pick some color and assign it to all of the vertices t_{ab}. Then, choose any K_4 gadget i, j, k_{ij}, t_{ij}

in H' corresponding to an edge ij in the original graph. Let $\hat{\imath}, \hat{\jmath}, \hat{k}_{ij}, \hat{t}_{ij}$ be the vectors assigned to the vertices, respectively. Assign three colors to the uncolored vertices arbitrarily. Then, choose any K_4 gadget $i, \ell, k_{i\ell}, t_{i\ell}$ adjacent to the first. For each uncolored vertex in the second clique, consider its assigned vector, determine which of $\hat{\imath}, \hat{\jmath}, \hat{k}_{ij}$ it is closet to, and assign the same color. We continue coloring adjacent cliques in this way, comparing vectors to the original set $\hat{\imath}, \hat{\jmath}, \hat{k}_{ij}$, until the coloring is propagated to the entire graph.

Although we assigned four colors, one of them is used exclusively by the vertices t_{ab}. Since these vertices are not in G, if this is a proper 4-coloring of H', then it gives a proper 3-coloring of the subgraph G. So, in the remainder of the proof we will show that no two adjacent vertices in H' were assigned the same color.

The graph H'' is comprised of m edge-disjoint gadgets R as well as m edges tt_{ab}. In order for the total objective value to be greater than $m\,\mathrm{MC}_3(K_3) + m - \varepsilon$, each gadget R must contribute at least $\mathrm{MC}_3(R) - \varepsilon$ and each edge tt_{ab} must contribute at least $1 - \varepsilon$. As shown in Lemma 31, the sum of squared edge lengths of a "tetrahedron with adjoined triangles" is nearly maximized iff the lengths of the edges forming the tetrahedron are nearly regular. Specifically, for a gadget R to achieve at least $10 + 2\sqrt{3} - \varepsilon$, each side of the tetrahedron must be in the interval $4/\sqrt{6} \pm \mathrm{O}(\sqrt{\varepsilon})$.

Now consider any two adjacent vertices v, u in H'. The two vertices exist in some K_4 gadget in H', and the coloring procedure must have reached it after at most m rounds. At each step in the procedure, there is a colored clique and a successive adjacent clique. There are two copies of R in H'' which contain those two 4-cliques from H'. The gadgets share one vertex. Additionally, each gadget has a vertex adjacent to the sink t, whose assigned vectors must be at distance at least $2 - 2\varepsilon$ from $\hat{t}$. Since all of these are in the unit sphere, standard trigonometry shows those two assigned vectors must themselves be within $\mathrm{O}(\varepsilon)$ of each other. As verified in Lemma 30, considering two nearly regular tetrahedra which share one vertex and have a pair of similar vertices, the other vertices of the tetrahedra must each be within $\mathrm{O}(\sqrt{\varepsilon})$ of each other.

After m rounds of the coloring procedure, we conclude that $\hat{v}, \hat{u}$ are each at distance at most $m \times \mathrm{O}(\sqrt{\varepsilon})$ away from the same-colored vectors in the original set $\hat{\imath}, \hat{\jmath}, \hat{k}_{ij}$. Since all of the edges of the 4-clique gadgets must be at least $4/\sqrt{6} - \mathrm{O}(\sqrt{\varepsilon})$, for $\varepsilon = \Theta(1/m^2)$ sufficiently small the vectors $\hat{v}, \hat{u}$ cannot be near the same original vector. Therefore, they must been colored differently, as desired. ◀

By the correspondence described earlier in the section, we immediately have the desired bound on $\mathcal{S}$-prodLH.

▶ **Corollary 6.** Quantum Max-Cut *restricted to product states,* prodQMC*, is* NP*-complete, even when all terms are restricted to have positive unit weight.*

References

1. Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.
2. Afonso S. Bandeira, Christopher Kennedy, and Amit Singer. Approximating the little Grothendieck problem over the orthogonal and unitary groups. *Mathematical programming*, 160:433–475, 2016. doi:10.1007/s10107-016-0993-7.
3. Alexander I. Barvinok. Problems of distance geometry and convex properties of quadratic maps. *Discrete & Computational Geometry*, 13:189–202, 1995. doi:10.1007/BF02574037.
4. Fernando G.S.L. Brandão and Aram W. Harrow. Product-state approximations to quantum states. *Commun. Math. Phys.*, 342:47–80, 2016. doi:10.1007/s00220-016-2575-1.

5 Sergey Bravyi, David Gosset, Robert König, and Kristan Temme. Approximation algorithms for quantum many-body problems. *Journal of Mathematical Physics*, 60(3):032203, 2019. doi:10.1063/1.5085428.

6 Sergey Bravyi and Matthew Hastings. On complexity of the quantum Ising model. *Communications in Mathematical Physics*, 349(1):1–45, 2017. doi:10.1007/s00220-016-2787-4.

7 Jop Briët, Harry Buhrman, and Ben Toner. A generalized Grothendieck inequality and nonlocal correlations that require high entanglement. *Communications in mathematical physics*, 305(3):827–843, 2011. doi:10.1007/s00220-011-1280-3.

8 Jop Briët, Fernando Mário de Oliveira Filho, and Frank Vallentin. The positive semidefinite Grothendieck problem with rank constraint. In *International Colloquium on Automata, Languages, and Programming*, pages 31–42. Springer, 2010. doi:10.1007/978-3-642-14165-2_4.

9 Jop Briët, Oded Regev, and Rishi Saket. Tight hardness of the non-commutative Grothendieck problem. *Theory of Computing*, 13(15):1–24, 2017. doi:10.4086/toc.2017.v013a015.

10 Anne Broadbent and Alex Bredariol Grilo. QMA-hardness of consistency of local density matrices with applications to quantum zero-knowledge. *SIAM Journal on Computing*, 51(4):1400–1450, 2022. doi:10.1137/21M140729X.

11 Samuel Burer and Renato D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical programming*, 95(2):329–357, 2003. doi:10.1007/s10107-002-0352-8.

12 Nadia Creignou. A dichotomy theorem for maximum generalized satisfiability problems. *Journal of Computer and System Sciences*, 51(3):511–522, 1995. doi:10.1006/jcss.1995.1087.

13 Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. *Complexity classifications of Boolean constraint satisfaction problems*. SIAM, 2001.

14 Toby Cubitt and Ashley Montanaro. Complexity classification of local Hamiltonian problems. *SIAM Journal on Computing*, 45(2):268–316, 2016. doi:10.1137/140998287.

15 Toby S. Cubitt, Ashley Montanaro, and Stephen Piddock. Universal quantum Hamiltonians. *Proceedings of the National Academy of Sciences*, 115(38):9497–9502, August 2018. doi:10.1073/pnas.1804949115.

16 Sevag Gharibian, Yichen Huang, Zeph Landau, Seung Woo Shin, et al. Quantum Hamiltonian complexity. *Foundations and Trends in Theoretical Computer Science*, 10(3):159–282, 2015. doi:10.1561/0400000066.

17 Sevag Gharibian and Julia Kempe. Approximation algorithms for QMA-complete problems. *SIAM Journal on Computing*, 41(4):1028–1050, 2012. doi:10.1137/110842272.

18 Sevag Gharibian and Ojas Parekh. Almost Optimal Classical Approximation Algorithms for a Quantum Generalization of Max-Cut. In *Approximation, Randomization, and Combinatorial Optimization (APPROX/RANDOM 2019)*, volume 145 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.APPROX-RANDOM.2019.31.

19 Michel X. Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995. doi:10.1145/227683.227684.

20 Sean Hallgren, Eunou Lee, and Ojas Parekh. An approximation algorithm for the max-2-local Hamiltonian problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.APPROX/RANDOM.2020.59.

21 Ryszard Horodecki et al. Information-theoretic aspects of inseparability of mixed states. *Phys. Rev. A*, 54(3):1838–1843, 1996. doi:10.1103/PhysRevA.54.1838.

22 Yeongwoo Hwang, Joe Neeman, Ojas Parekh, Kevin Thompson, and John Wright. Unique games hardness of Quantum Max-Cut, and a conjectured vector-valued Borell's inequality. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1319–1384. SIAM, 2023. doi:10.1137/1.9781611977554.ch48.

23 Peter Jonsson. Boolean constraint satisfaction: complexity results for optimization problems with arbitrary weights. *Theoretical Computer Science*, 244(1-2):189–203, 2000. doi:10.1016/S0304-3975(98)00343-0.

24 Peter Jonsson, Mikael Klasson, and Andrei Krokhin. The approximability of three-valued Max CSP. *SIAM Journal on Computing*, 35(6):1329–1349, 2006. doi:10.1137/S009753970444644X.

25 Sanjeev Khanna, Madhu Sudan, and David P Williamson. A complete classification of the approximability of maximization problems derived from boolean constraint satisfaction. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 11–20, 1997. doi:10.1145/258533.258538.

26 Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and quantum computation*. Number 47 in Graduate Studies in Mathematics. American Mathematical Soc., 2002.

27 Yi-Kai Liu. Consistency of local density matrices is QMA-complete. In Josep Díaz, Klaus Jansen, José D. P. Rolim, and Uri Zwick, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 438–449, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. doi:10.1007/11830924_40.

28 L. Lovász. *Semidefinite Programs and Combinatorial Optimization*, pages 137–194. Springer New York, New York, NY, 2003. doi:10.1007/0-387-22444-0_6.

29 László Lovász. *Graphs and geometry*, volume 65. American Mathematical Soc., 2019.

30 Hiroshi Maehara. On the total edge-length of a tetrahedron. *The American Mathematical Monthly*, 108(10):967–969, 2001. URL: http://www.jstor.org/stable/2695418.

31 Ojas Parekh and Kevin Thompson. Beating random assignment for approximating quantum 2-local Hamiltonian problems. In *29th Annual European Symposium on Algorithms (ESA 2021)*, volume 204 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 74:1–74:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ESA.2021.74.

32 Ojas Parekh and Kevin Thompson. An optimal product-state approximation for 2-local quantum Hamiltonians with positive terms, 2022. arXiv:2206.08342v1.

33 Stephen Piddock and Ashley Montanaro. The complexity of antiferromagnetic interactions and 2D lattices. *Quantum Info. Comput.*, 17(7-8):636–672, 2017. doi:10.5555/3179553.3179559.

34 Stephen Piddock and Ashley Montanaro. Universal qudit Hamiltonians, 2018. arXiv:1802.07130.

35 Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 216–226, 1978. doi:10.1145/800133.804350.

36 Johan Thapper and Stanislav Živnỳ. The complexity of finite-valued CSPs. *Journal of the ACM*, 63(4):1–33, 2016. doi:10.1145/2974019.

37 G. Thompson. Normal forms for skew-symmetric matrices and Hamiltonian systems with first integrals linear in momenta. *Proc. of the Amer. Math. Soc.*, 104(3):910–916, 1988. doi:10.2307/2046815.

38 John Watrous. *The theory of quantum information*. Cambridge university press, 2018.

39 Pawel Wocjan and Thomas Beth. The 2-local Hamiltonian problem encompasses NP. *International Journal of Quantum Information*, 1(03):349–357, 2003. doi:10.1142/S021974990300022X.

40 John Wright. Personal communication, 2022.

A Geometry Lemmas

First, we give a simple lemma regarding star graphs. Our goal is to show that inserting star gadgets into a graph forces maximal solutions to $\mathrm{MC}^{\mathrm{L}}_W$ to be close to the highest weighted axes.

▶ **Lemma 26.** *Consider a star graph S_K with center vertex v and K neighbors. Consider any $0 \leq \varepsilon \leq 1$ and any $W = \mathrm{diag}(1, w_2, w_3)$ with $1 \geq w_2 \geq w_3 \geq 0$ and $w_3 < 1$.*

Let Π *denote the projector onto the axes for which* $w_i = 1$ *(so the projector onto the* x*-axis or the* xy*-plane). Similarly, let* λ *be the largest* w_i *which is not equal to 1 (either* w_2 *or* w_3*). Let* $\widehat{\operatorname{sgn}}(\hat{v}) = \Pi v / \|\Pi v\|$.

If vectors are assigned to S_K *which achieve at least* $\mathrm{MC}^{\mathrm{L}}_{\mathrm{W}}(S_K) \geq K(1-\varepsilon)$*, and* $\hat{v}$ *is the vector assigned to* v*, then*

$$\|W\hat{v} - \widehat{\operatorname{sgn}}(\hat{v})\| \leq \delta \quad \textit{for} \quad \delta = 2\sqrt{\varepsilon}\sqrt{\frac{1+\lambda^2}{1-\lambda^2}}.$$

Proof. Suppose to the contrary that $\|W\hat{v} - \widehat{\operatorname{sgn}}(\hat{v})\| > \delta$. We know that

$$\begin{aligned}
&\|W\hat{v} - \widehat{\operatorname{sgn}}(\hat{v})\| = \sqrt{(\widehat{\operatorname{sgn}}(\hat{v})_x - \hat{v}_x)^2 + (\widehat{\operatorname{sgn}}(\hat{v})_y - w_2\hat{v}_y)^2 + w_3^2\hat{v}_z^2} \\
&= \sqrt{(\widehat{\operatorname{sgn}}(\hat{v})_x - \hat{v}_x)^2 + (\widehat{\operatorname{sgn}}(\hat{v})_y - w_2\hat{v}_y)^2 + w_3^2\left(1 - \hat{v}_x^2 - \hat{v}_y^2\right)} \\
&\leq \sqrt{\left(1 - \hat{v}_x^2 - \hat{v}_y^2\right) + \lambda^2(1 - \hat{v}_x^2 - \hat{v}_y^2)} = \sqrt{(1+\lambda^2)(1 - \hat{v}_x^2 - \hat{v}_y^2)},
\end{aligned}$$

where the inequality holds for $\hat{v}_x, \hat{v}_y \in [0,1]$. Combining, we have that $(1+\lambda^2)(1-\hat{v}_x^2-\hat{v}_y^2) > \delta^2$, which implies $\hat{v}_x^2 + \hat{v}_y^2 < 1 - \delta^2/(1+\lambda^2)$. Now we are ready to find that the maximum objective value earned on the star is less than half of

$$\begin{aligned}
&K(1 + \|W\hat{v}\|) = K + K\sqrt{\hat{v}_x^2 + \hat{v}_y^2 + \lambda^2\hat{v}_z^2} = K + K\sqrt{\hat{v}_x^2 + \hat{v}_y^2 + \lambda^2(1 - \hat{v}_x^2 - \hat{v}_y^2)} \\
&= K + K\sqrt{(\hat{v}_x^2 + \hat{v}_y^2)(1-\lambda^2) + \lambda^2} \leq K + K\sqrt{\left(1 - \frac{\delta^2}{1+\lambda^2}\right)(1-\lambda^2) + \lambda^2} \\
&= K + K\sqrt{1 - 4\varepsilon} \leq K + K(1 - 4\varepsilon) = K(2 - 4\varepsilon).
\end{aligned}$$

So, the objective value is less than $K(1-2\varepsilon)$, which is less than $K(1-\varepsilon)$, a contradiction. ◀

Next, we study the geometry of triangles. For a triangle ABC, let $\mathrm{L}(ABC)$ denote the sum of the edge lengths. It is straightforward to show that for a triangle inscribed in a circle of radius r, $\mathrm{L}(ABC) \leq 3\sqrt{3}r$, which is uniquely achieved by an equilateral triangle [30].

▶ **Lemma 27.** *Consider a triangle* ABC *inscribed in the unit circle and any* $0 \leq \varepsilon < 1$*. If* $\mathrm{L}(ABC) \geq 3\sqrt{3} - \varepsilon$*, then each edge length is in the interval* $\left[\sqrt{3} - 4\sqrt{\varepsilon}, \sqrt{3} + 4\sqrt{\varepsilon}\right]$.

Proof. For the sake of contradiction, suppose $\mathrm{L}(ABC) \geq 3\sqrt{3} - \varepsilon$ and that there exists an edge length outside the interval $\sqrt{3} \pm 4\sqrt{\varepsilon}$.

First suppose that some edge length is greater than $\sqrt{3} + 4\sqrt{\varepsilon}$. Because the maximum value of L is $3\sqrt{3}$, this implies at least one edge length is less than $\sqrt{3} - 2\sqrt{\varepsilon}$. So whether an edge length is above or below the interval, some edge length is less than $\sqrt{3} - 2\sqrt{\varepsilon}$.

Given $\mathrm{L}(ABC) \geq 3\sqrt{3} - \varepsilon$ and some edge length is less than $\sqrt{3} - 2\sqrt{\varepsilon}$, some edge length must be greater than $(3\sqrt{3} - \varepsilon - \sqrt{3} + 2\sqrt{\varepsilon})/2 \geq \sqrt{3} + \sqrt{\varepsilon}$, so there exists a pair of edge lengths whose difference is greater than $3\sqrt{\varepsilon}$.

We relabel the triangle so that $\|AC\| \geq \|AB\| \geq \|BC\|$ and $\|AC\| - \|BC\| > 3\sqrt{\varepsilon}$. We can rotate the unit circle as necessary so that ABC is of the form

$$A = (a, b) \quad B = (-a, b) \quad C = (c_x, c_y).$$

Define $C' = (0, -1)$. We will show $\mathrm{L}(ABC') - \mathrm{L}(ABC) > \varepsilon$, implying $\mathrm{L}(ABC)$ is less than $3\sqrt{3} - \varepsilon$, a contradiction.

Both $\mathrm{L}(ABC)$ and $\mathrm{L}(ABC')$ are sums of 3 edge lengths. The edge AB is shared and we have that $\|AC'\| = \|BC'\|$, so we are interested in $2\|AC'\| - \|BC\| - \|AC\|$. We may calculate

$$\|AC'\| = \sqrt{2+2b}, \quad \|BC\| = \sqrt{2+2bc_y+2ac_x}, \quad \|AC\| = \sqrt{2+2bc_y-2ac_x}.$$

From here, we may verify

$$4\|AC'\|^2 = (\|BC\| + \|AC\|)^2 + (\|BC\| - \|AC\|)^2,$$

and so

$$(2\|AC'\| + \|BC\| + \|AC\|)(2\|AC'\| - \|BC\| - \|AC\|) \geq (\|BC\| - \|AC\|)^2,$$

and

$$2\|AC'\| - \|BC\| - \|AC\| \geq (\|BC\| - \|AC\|)^2 (2\|AC'\| + \|BC\| + \|AC\|)^{-1}.$$

To lower bound the left-hand side, we use our lower bound on the difference between the edge lengths and the upper bound that all lengths in the unit circle are less than 2. We find

$$2\|AC'\| - \|BC\| - \|AC\| > \left(3\sqrt{\varepsilon}\right)^2 8^{-1} = 9\varepsilon/8.$$

Overall, we have found that $\mathrm{L}(ABC') - \mathrm{L}(ABC) > \varepsilon$. ◀

▶ **Lemma 28.** *Consider triangles ABC and ADE inscribed in the unit circle. If all edges of the triangle have lengths in the interval $\sqrt{3} \pm \delta$, then the points $\{B, C\}$ are each within $\mathrm{O}(\delta)$ distance of (different) points in $\{D, E\}$.*

Proof. Because these are vectors restricted to a unit circle, we can assume $A = (1, 0, 0)$. Given A is fixed, we can characterize the constraints on the other points as follows. B, C, D, E must lie in the intersection of the unit circle with a circular shell bounded by radii $\sqrt{3} - \delta$ and $\sqrt{3} + \delta$ centered at A. Given $\sqrt{3} + \delta < 2$, it is clear this intersection is two disjoint segments of the unit circle.

We wish to upper bound the distance between points in either of these regions. This is the distance between the ends of the regions, along a chord from the point at distance $\sqrt{3} - \delta$ to the point at distance $\sqrt{3} + \delta$.

The length of the chord can be bound as follows. Given a chord length d, the internal angle is $2\arcsin(d)$. Given an angle θ, the chord length is $2\sin(\theta/2)$. So, we can convert the known distances to angles, take the difference, and convert back to a distance: $2\sin(\arcsin((\sqrt{3} + \delta)/2) - \arcsin((\sqrt{3} - \delta)/2))$. This is equivalent to $2\sin\left(\arcsin(d/2) - \arcsin\left((d - 2\delta)/2\right)\right)$ for $d = \sqrt{3} + \delta$. Observing this function is increasing in d, we can upper bound the value by taking $d < \sqrt{3} + 0.1$. The Taylor series of $2\sin(\arcsin(\frac{\sqrt{3}+0.1}{2}) - \arcsin(\frac{\sqrt{3}+0.1-\delta}{2}))$ gives that this is at most 2.5δ. ◀

Next, we transition to considering tetrahedra. For a tetrahedron $ABCD$, let $\mathrm{L}(ABCD)$ denote the sum of the edge lengths. It is known that for a tetrahedron inscribed in a sphere of radius r, $\mathrm{L}(ABCD) \leq 4\sqrt{6}r$, and the maximum is uniquely achieved by a regular tetrahedron [30].

▶ **Lemma 29.** *Consider a tetrahedron $ABCD$ inscribed in the unit sphere and any $\varepsilon \geq 0$. If $\mathrm{L}(ABCD) \geq 4\sqrt{6} - \varepsilon$, then each edge length is in the interval $\left[\frac{4\sqrt{6}}{6} - \frac{50\sqrt{2\varepsilon}}{3}, \frac{4\sqrt{6}}{6} + \frac{50\sqrt{2\varepsilon}}{3}\right]$.*

Proof. For the sake of contradiction, suppose $\mathrm{L}(ABCD) \geq 4\sqrt{6} - \varepsilon$ and that there exists an edge length outside the given interval.

First consider the case that some edge length is greater than $\frac{4\sqrt{6}}{6} + \frac{50\sqrt{2\varepsilon}}{3}$. Because the maximum of L in the unit sphere is $4\sqrt{6}$, at least one of the other edge lengths must be less than $\frac{4\sqrt{6}}{6} - \frac{10\sqrt{2\varepsilon}}{3}$. So, whether an edge length is above or below the interval, in both cases some edge length is less than $\frac{4\sqrt{6}}{6} - \frac{10\sqrt{2\varepsilon}}{3}$.

If $\mathrm{L}(ABCD) \geq 4\sqrt{6} - \varepsilon$ and some edge length is less than $\frac{4\sqrt{6}}{6} - \frac{10\sqrt{2\varepsilon}}{3}$, then some edge length must be greater than $\left(4\sqrt{6} - \varepsilon - \frac{4\sqrt{6}}{6} + \frac{10\sqrt{2\varepsilon}}{3}\right)/5 \geq \frac{4\sqrt{6}}{6} + \frac{2\sqrt{2\varepsilon}}{3}$. So, there exists a pair of edges such that the difference in their lengths is greater than $\frac{12\sqrt{2\varepsilon}}{3} = 4\sqrt{2\varepsilon}$.

If two edge lengths e_1, e_2 in a tetrahedron differ by more than $4\sqrt{2\varepsilon}$, then there must exist a pair of adjacent edges which differ by more than $2\sqrt{2\varepsilon}$. This is because either the two edges are adjacent, or they are both adjacent to a third edge e_3 and that length must be closer to e_1 or to e_2, i.e. $\min\{\|e_3 - e_1\|, \|e_2 - e_1\|\} \leq \|e_1 - e_2\|/2$ We relabel the tetrahedron so that $\|AC\| - \|BC\| > 2\sqrt{2\varepsilon}$.

Next, without changing the value of L, we can rotate $ABCD$ such that A, B are of the form $A = (a, -b, 0) \quad B = (-a, -b, 0)$. Let $C = (c_x, c_y, c_z) \quad D = (d_x, d_y, d_z)$ and $C' = (0, c_y, \sqrt{1 - c_y^2}) \quad D' = (0, d_y, \sqrt{1 - d_y^2})$, We will show $\mathrm{L}(ABC'D') - \mathrm{L}(ABCD) > \varepsilon$, implying $\mathrm{L}(ABCD)$ must in fact be less than $4\sqrt{6} - \varepsilon$.

Both $\mathrm{L}(ABCD)$ and $\mathrm{L}(ABC'D')$ are sums of 6 edge lengths. We can ignore AB. We can directly verify that $\|C'D'\| - \|CD\|$ is

$$\sqrt{(c_y - d_y)^2 + (c_x - d_x)^2 + (c_z - d_z)^2 + 2c_x d_x + 2c_z d_z + 2\sqrt{c_x^2 + c_z^2}\sqrt{d_x + d_z^2}} - \sqrt{(c_x - d_x)^2 + (c_y - d_y)^2 + (c_z - d_z)^2},$$

and so is clearly non-negative. We may calculate

$$\|AC'\| = \|BC'\| = \sqrt{2 + 2bc_y}$$
$$\|BC\| = \sqrt{2 + 2bc_y + 2ac_x}$$
$$\|AC\| = \sqrt{2 + 2bc_y - 2ac_x}$$

and similar expressions for $\|AD'\|, \|BD'\|, \|BD\|, \|AD\|$.

Just as in the proof of Lemma 27, we may verify

$$\begin{aligned} 2\|AC'\| - \|BC\| - \|AC\| &= (\|BC\| - \|AC\|)^2 \left(2\|AC'\| + \|BC\| + \|AC\|\right)^{-1} \\ &> \left(2\sqrt{2\varepsilon}\right)^2 8^{-1} \\ &= \varepsilon. \end{aligned}$$

Similarly, $2\|AD'\| - \|AD\| - \|BD\|$ is twice the difference of the quadratic mean and arithmetic mean of $\|AD\|, \|BD\|$, which is non-negative by the generalized mean inequality.

Overall, we have found that $\mathrm{L}(ABC'D') - \mathrm{L}(ABCD) > \varepsilon$. ◀

▶ **Lemma 30.** *Consider tetrahedra $ABCD$ and $AEFG$ inscribed in the unit sphere with $\|B - E\| \leq \varepsilon$. If all edges of the tetrahedra have lengths in the interval $\frac{4\sqrt{6}}{6} \pm \delta$, then the points $\{C, D\}$ are each within $\mathrm{O}(\delta + \varepsilon)$ distance of (different) points in $\{F, G\}$.*

Proof. Because this is the unit sphere, we can rotate so that A, B are in the xy-plane and equidistant from the y-axis.

Given A, B, E are fixed, we can characterize the constraints on the other points as follows. Let R_1 denote the region bounded by a spherical shell centered at A with radii $\frac{4\sqrt{6}}{6} - \delta$ and $\frac{4\sqrt{6}}{6} + \delta$. Similarly, let R_2 denote a region of the same size and shape but centered at B. The points C, D must exist in the intersection of R_1, R_2, and the unit sphere. The constraints imply the intersection is two disjoint sectors of the unit sphere opposite A and B, call them S_1, S_2. Similarly, the points F, G must exist within ε of S_1 and S_2.

We wish to show that any two points in S_1 must be close, and similarly for S_2. Consider any two points P_1, P_2 in S_1. Fixing P_1 and letting $d(P_2) = \|P_1 - P_2\|$, the function d is convex in P_2 over S_1. Therefore, the extremes of the distance $\|P_1 - P_2\|$ will occur at the four extremal points of S_1, identified by their distances $\frac{4\sqrt{6}}{6} \pm \delta$ from A and B.

If P_1 is at $\frac{4\sqrt{6}}{6} + \delta$ from A and from B, then $d(P_2)$ is maximized with P_2 at $\frac{4\sqrt{6}}{6} - \delta$ from A and B. From point P_1, moving 2δ towards A and then 2δ towards B is an upper bound on the distance to P_2, so $\|P_1 - P_2\| \leq 4\delta$.

The other extreme may occur with $\frac{4\sqrt{6}}{6} + \delta$ and $\frac{4\sqrt{6}}{6} - \delta$ as the distances from P_1 to A and B, respectively, and as the distances from P_2 to B and A, respectively. Recalling that we were able to assume A, B are in the xy-plane and symmetric about the z-axis, it is clear that P_1, P_2 must be symmetric such that $P_1 = (0, P_y, P_z), P_2 = (0, -P_y, P_z)$. This implies A, B, P_1, P_2 are coplanar, and so ABP_1P_2 is a cyclic quadrilateral inscribed in a circular cross-section of the unit sphere. We know the lengths of three sides and the diagonals of this quadrilateral and want to know the fourth side, $\|P_1P_2\|$. Ptolmey's Theorem for cyclic quadrilaterals gives us that

$$\|AP_1\| \times \|BP_2\| = \|AB\| \times \|P_1P_2\| + \|AP_2\| \times \|BP_1\|.$$

Simplifying, we find

$$\|P_1P_2\| \leq \frac{\left(4\sqrt{6} + \delta\right)^2 - \left(4\sqrt{6} - \delta\right)^2}{4\sqrt{6} - \delta} = \frac{8\sqrt{6}\delta}{2\sqrt{6} - 3\delta} \leq \frac{8\sqrt{6}\delta}{2\sqrt{6}} = 4\delta.$$

Overall, we conclude that given any two points in S_1, the distance between the points is at most $\mathrm{O}(\delta)$, and similarly for S_2.

Finally, we return to the shapes $ABCD$ and $AEFG$. Since the distance $\|CD\|$ must be in $\frac{4\sqrt{6}}{6} \pm \delta$, one point must be in S_1 and one point in S_2. Similarly, one of F, G must be near S_1 and one near S_2. So arbitrarily, C, F are within ε of S_1 and D, G are within ε of S_2. We can conclude $\|C - F\|$ and $\|D - G\|$ are at most $\mathrm{O}(\delta + \varepsilon)$, as desired. ◀

We will use the name "tetrahedron with adjoined triangles" to refer to a three-dimensional tetrahedron such that for each edge there is an additional point with which the ends of the edge are joined to form a triangle. In total, there are 10 vertices and 12 edges. The shape will be assumed to be inscribed in the unit sphere, so a vector describing any vertex of the tetrahedron or a triangle is a unit vector. For some instance R of this shape with assigned vertex locations, let $\mathrm{L}(R)$ denote the sum of the *squared* edge lengths. The following lemma says that for $\mathrm{L}(R)$ to be maximized, the edges of the tetrahedron must approximate a regular tetrahedron.

▶ **Lemma 31.** *Given a tetrahedron with adjoined triangles R, $\mathrm{L}(R) \leq 10 + 2\sqrt{3}$. Furthermore, If any edge of the tetrahedron has a length outside of the interval $[\frac{4}{\sqrt{6}} - \varepsilon, \frac{4}{\sqrt{6}} + \varepsilon]$, then*

$$\mathrm{L}(R) < 10 + 2\sqrt{3} - \Omega\left(\varepsilon^2\right).$$

Proof. Suppose the vertices of the tetrahedron are V_1, V_2, V_3, V_4, and the additional vertex forming a triangle with edge V_iV_j is v_{ij}. Let s_{ij} denote the length of edge ij.

Note that the optimal location (which maximizes L) of a vertex v_{ij} is entirely determined by the locations of V_i and V_j, since it has no other neighbors. Given V_i, V_j, the point which maximizes the squared distances is proportional to $-(V_i + V_j)$. Moreover, because we are in the unit sphere, the maximum objective achieved by a triangle $V_iV_jv_{ij}$ when V_i, V_j are fixed is entirely determined by the length of V_iV_j alone; their exact coordinates do not matter, due to rotational symmetry. Combining these two facts and applying trigonometry, we may calculate that given an edge length of the tetrahedron s_{ij}, the maximum objective value of the triangle $V_iV_jv_{ij}$ is

$$\mathrm{t}(s_{ij}) = 1 + \frac{s_{ij}^2}{4} + \sqrt{1 - \frac{s_{ij}^2}{4}}.$$

Now, because R is the disjoint union of the six triangles which each have one of the tetrahedron's edges as a side, we can express

$$\mathrm{L}(R) = \sum_{i<j}^{4} t(s_{ij}).$$

A regular tetrahedron inscribed in the unit sphere has edge lengths equal to $4/\sqrt{6}$. So, if R is formed by a regular tetrahedron and the ideal points for the triangles, then $\mathrm{L}(R) = 6\,\mathrm{t}(4/\sqrt{6}) = 10 + 2\sqrt{3}$.

Now, we suppose the tetrahedron is not regular and some side length is not in the interval $\frac{4}{\sqrt{6}} \pm \varepsilon$. As noted in prior lemmas, it is known that the maximum sum of the (unsquared) edge-lengths of a tetrahedron is $4\sqrt{6}$ [30], so we have $\sum_{i<j} s_{ij} \leq 4\sqrt{6}$. If any edge length is greater than $\frac{4}{\sqrt{6}} + \varepsilon$, then the sum of the other five sides must be greater $5 \times \frac{4}{\sqrt{6}} - \varepsilon$ (otherwise, $\mathrm{L}(R)$ could be trivially increased, meaning our assumption gives a lower bound on the difference), so some side is less than $\frac{4}{\sqrt{6}} - \varepsilon/5$. So whether some side is above or below the interval, there exists a side $s_{ij} < \frac{4}{\sqrt{6}} - \varepsilon/5$.

Analyzing $\mathrm{t}(s)$, we see it is increasing and concave down on the interval $[1.22, 2)$. The maximum occurs at $\mathrm{t}(\sqrt{3})$. We can assume all edge lengths are at least 1.22 and in the concave-down region, as otherwise the total edge length is less than $\mathrm{t}(1.22)+5\,\mathrm{t}(\sqrt{3}) = 13.4145$, which is bounded away from the optimum $10 + 2\sqrt{3} \approx 13.464$. Therefore, the sum of t over the edges of a regular tetrahedron are greater than over a non-regular tetrahedron:

$$6\,\mathrm{t}\left(\frac{4}{\sqrt{6}}\right) > \sum_{i<j}^{4} \mathrm{t}(s_{ij}).$$

In particular,

$$6\,\mathrm{t}\left(\frac{4}{\sqrt{6}}\right) - \sum_{i<j}^{4} \mathrm{t}(s_{ij}) \geq 2\,\mathrm{t}\left(\frac{4}{\sqrt{6}}\right) - \mathrm{t}\left(\frac{4}{\sqrt{6}} - \frac{\varepsilon}{5}\right) - \mathrm{t}\left(\frac{4}{\sqrt{6}} + \frac{\varepsilon}{5}\right).$$

After extensive calculation, the difference of the sum over a regular tetrahedron (the LHS) minus the sum over the non-regular tetrahedron (the RHS) is $\Omega(\varepsilon^2)$. ◀

A Universal Sequence of Tensors for the Asymptotic Rank Conjecture

Petteri Kaski
Department of Computer Science, Aalto University, Finland

Mateusz Michałek
Fachbereich Mathematik und Statistik, Universität Konstanz, Germany

Abstract

The *exponent* $\sigma(T)$ of a tensor $T \in \mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d$ over a field $\mathbb{F}$ captures the base of the exponential growth rate of the tensor rank of T under Kronecker powers. Tensor exponents are fundamental from the standpoint of algorithms and computational complexity theory; for example, the exponent ω of square matrix multiplication can be characterized as $\omega = 2\sigma(\mathrm{MM}_2)$, where $\mathrm{MM}_2 \in \mathbb{F}^4 \otimes \mathbb{F}^4 \otimes \mathbb{F}^4$ is the tensor that represents 2×2 matrix multiplication.

Strassen [FOCS 1986] initiated a duality theory for spaces of tensors that enables one to characterize the exponent of a tensor via objects in a dual space, called the *asymptotic spectrum* of the primal (tensor) space. While Strassen's theory has considerable generality beyond the setting of tensors – Wigderson and Zuiddam [*Asymptotic Spectra: Theory, Applications, and Extensions*, preprint, 2023] give a recent exposition – progress in characterizing the dual space in the tensor setting has been slow, with the first universal points in the dual identified by Christandl, Vrana, and Zuiddam [*J. Amer. Math. Soc.* 36 (2023)]. In parallel to Strassen's theory, the algebraic geometry community has developed a geometric theory of tensors aimed at characterizing the structure of the primal space and tensor exponents therein; the latter study was motivated in particular by an observation of Strassen (implicit in [*J. Reine Angew. Math.* 384 (1988)]) that matrix-multiplication tensors have limited universality in the sense that $\sigma(\mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d) \leq \frac{2\omega}{3} = \frac{4}{3}\sigma(\mathrm{MM}_2)$ holds for all $d \geq 1$. In particular, this limited universality of the tensor MM_2 puts forth the question whether one could construct explicit universal tensors that exactly characterize the worst-case tensor exponent in the primal space. Such explicit universal objects would, among others, give means towards a proof or a disproof of Strassen's asymptotic rank conjecture [*Progr. Math.* 120 (1994)]; the former would immediately imply $\omega = 2$ and, among others, refute the Set Cover Conjecture (cf. Björklund and Kaski [STOC 2024] and Pratt [STOC 2024]).

Our main result is an explicit construction of a *sequence* $\mathcal{U}_d$ of zero-one-valued tensors that is universal for the worst-case tensor exponent; more precisely, we show that $\sigma(\mathcal{U}_d) = \sigma(d)$ where $\sigma(d) = \sup_{T \in \mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d} \sigma(T)$. We also supply an explicit universal sequence $\mathcal{U}_\Delta$ localised to capture the worst-case exponent $\sigma(\Delta)$ of tensors with support contained in $\Delta \subseteq [d] \times [d] \times [d]$; by combining such sequences, we obtain a universal sequence $\mathcal{T}_d$ such that $\sigma(\mathcal{T}_d) = 1$ holds if and only if Strassen's asymptotic rank conjecture holds for d. Finally, we show that the limit $\lim_{d\to\infty} \sigma(d)$ exists and can be captured as $\lim_{d\to\infty} \sigma(D_d)$ for an explicit sequence $(D_d)_{d=1}^{\infty}$ of tensors obtained by diagonalisation of the sequences $\mathcal{U}_d$.

As our second result we relate the *absence* of polynomials of fixed degree vanishing on tensors of low rank, or more generally asymptotic rank, with upper bounds on the exponent $\sigma(d)$. Using this technique, one may bound asymptotic rank for all tensors of a given format, knowing enough specific tensors of low asymptotic rank.

2012 ACM Subject Classification Mathematics of computing; Theory of computation → Algebraic complexity theory

Keywords and phrases asymptotic rank conjecture, secant variety, Specht module, tensor rank, tensor exponent

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.64

Related Version *Previous Version*: https://arxiv.org/abs/2404.06427



© Petteri Kaski and Mateusz Michałek;
licensed under Creative Commons License CC-BY 4.0



Funding *Mateusz Michałek*: Supported by the DFG grant 467575307.

Acknowledgements We would like to thank Andreas Björklund, Joseph Landsberg, and Peter Vrana for useful discussions and important comments.

1 Introduction

1.1 Exponents of Tensors and the Quest for Universality

For an infinite field $\mathbb{F}$ and a positive integer constant d, the *exponent* of a nonzero tensor $T \in \mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d$ is the least nonnegative real $\sigma(T)$ such that the sequence of tensor (Kronecker) powers

$$T^{\boxtimes \mathbb{N}} = (T^{\boxtimes p} : p = 1, 2, \ldots) \tag{1}$$

has its sequence of tensor ranks bounded by $\mathrm{R}(T^{\boxtimes p}) \leq d^{\sigma(T)p+o(p)}$. Exponents of specific *constant-size* tensors T are of fundamental significance in the study of algorithms and algebraic complexity theory [16, 76]. For example, Strassen showed that the exponent ω of square matrix multiplication satisfies $\omega = 2\sigma(\mathrm{MM}_2)$ [71, 67], where MM_2 is the $4 \times 4 \times 4$ tensor representing the multiplication map for two 2×2 matrices. Recently, it was shown that the Set Cover Conjecture [29, 30, 40] fails if the exponent of a specific $7 \times 7 \times 7$ tensor Q is sufficiently close to one [7, 59].

The study of exponents of tensors – or, what is the same up to exponentiation, the study of *asymptotic rank*[1] [34] of tensors – is difficult. The exponent ω of square matrix multiplication is perhaps the best studied nontrivial exponent, and even in its case the best current lower bound $\omega \geq 2$ remains the trivial one (but see [8, 43, 52, 47, 24]), and the best current upper bound $\omega \leq 2.371866$ [31] is a result of extensive work spanning decades (e.g. [70, 58, 6, 62, 61, 25, 71, 26, 65, 75, 53, 2]) and relying on increasingly sophisticated techniques. In parallel to the study of exponents of individual tensors, research effort has been invested into developing a structural theory for spaces of tensors and their exponents, a theory to which the present paper also contributes. Two rough lines of research most relevant to our present work are as follows.

Strassen's duality theory – the asymptotic spectrum of tensors. The first line of research, announced by Strassen [71] in his 1986 FOCS paper and developed in a sequence of papers [66, 67, 68] and PhD theses [15, 74, 54] (cf. [72]), builds a duality theory for asymptotic rank of tensors based on the theory of preordered commutative semirings, with the direct sum and tensor (Kronecker) product of tensors as the pertinent semiring operations, and the preorder defined by a rank-capturing tensor restriction relation. Wigderson and Zuiddam [76] give a recent comprehensive exposition of Strassen's theory and the preorder-theoretic topological dual spaces – the *asymptotic spectrum* of a space of tensors – which enable a tight characterization of the asymptotic rank of a tensor via the preorder-monotone homomorphisms in the dual. Yet, progress in terms of understanding the structure and identifying explicit points in the asymptotic spectrum of tensors has been slow. Beyond Strassen's original construction of support functionals, which are restricted to the subspace of oblique tensors, only recently Christandl, Vrana, and Zuiddam [18] constructed a family

[1] The *asymptotic rank* of $T \in \mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d$ is $\tilde{\mathrm{R}}(T) = \lim_{p\to\infty} \mathrm{R}(T^{\boxtimes p})^{1/p}$ and we have $\tilde{\mathrm{R}}(T) = d^{\sigma(T)}$.

of explicit *universal* spectral points – called quantum functionals – using theory of quantum entropy and covariants; however, also this dual family is far from yielding broadly tight lower bounds on asymptotic rank.

Viewed from the standpoint of Strassen's duality theory, rather than working in a dual space, in this paper we seek and obtain as our main result an explicit and universal *primal* characterization of tensor exponents and thus asymptotic rank in $\mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d$ – before proceeding to our results, let us review a second pertinent line of prior research and motivation.

Geometry of tensors and the asymptotic rank conjecture. The second line of research seeks to study spaces of tensors to identify the *worst-case* behaviour in the space using tools from algebraic geometry, in particular building on the seminal concept of *border rank* of a tensor due to Bini, Capovani, Romani, and Lotti [6] (see also Schönhage [62]) and its geometric characterization via secant varieties of Segre and Veronese varieties (e.g. [13, 14, 50, 42, 45, 78]). For a space $\mathcal{F}$ of tensors, let $\sigma(\mathcal{F}) = \sup_{T\in\mathcal{F}} \sigma(T)$; as a special case, for $d = 1, 2, \ldots$ let $\sigma(d) = \sigma(\mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d)$. It is immediate that $1 \leq \sigma(d) \leq 2$ and that $\sigma(1) = 1$. Over the complex numbers, it is a nontrivial consequence of the geometry of tensors that $\sigma(2) = 1$. Already the value $\sigma(3)$ is open and would be of substantial interest to determine. For example, it is known that $\sigma(3) = 1$ implies $\omega = 2$ by the application of the Coppersmith–Winograd method [26] to a specific $3 \times 3 \times 3$ tensor. It appears very difficult to prove lower bounds on $\sigma(d)$ apart from the trivial $\sigma(d) \geq 1$. Indeed, any tensor $T \in (\mathbb{F}^d)^{\otimes 3}$ with $\sigma(T) > 1$ would yield an explicit sequence of tensors, namely its Kronecker powers, such that for any constant $C > 0$ for $k \gg 0$ we have rank and border rank of $T^{\boxtimes k}$ greater than Cd^k. Currently, we do not know explicit examples of such sequences for $C = 3$ with the current world record lower bound on rank in [1]. One of the main obstacles is lack of methods to prove that a given tensor has high rank or border rank. In case of rank, the state of the art is the substitution method, based on linear algebra. In case of border rank, one looks for polynomial witnesses that vanish on tensors of given bounded border rank. However in this case, most known equations, not found via explicit computations, also vanish on so-called cactus varieties, which fill the ambient space quickly and thus known methods cannot provide super-linear lower bounds on tensor rank [4, 5, 12, 32, 38]. The state of the art for border rank is based on mixture of different methods, barely breaking the bound for $C = 2$ [48]. As the rank of the generic tensor grows quadratically with d the problem is often referred to as an instance of the *hay in a haystack* problem (phrase due to H. Karloff): find an explicit object that behaves generically. There is hope in the new introduced method of border apolarity [13], still there is currently no clear path of getting past $C = 3$.

The difficulty of lower bounds and progressively improving upper bounds for specific exponents, the exponent ω in particular, has prompted bold conjectures on worst-case exponents for broad families of tensors. Most notably, writing $\mathcal{T}_d$ for the space of all tight tensors in $\mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d$, *Strassen's asymptotic rank conjecture* (cf. [69, Conjecture 5.3]) states that the worst-case exponent for this space is the least possible:

▶ **Conjecture 1** (Strassen's asymptotic rank conjecture). *For all $d \geq 1$ it holds that $\sigma(\mathcal{T}_d) = 1$.*

Strassen's asymptotic rank conjecture, if true, immediately implies the algorithmically serendipitous corollary $\omega = 2$ in particular; cf. also [7, 59] for further consequences to algorithms. A yet stronger conjecture (cf. Bürgisser, Clausen, and Shokrollahi [16, Problem 15.5]; also e.g. Conner, Gesmundo, Landsberg, Ventura, and Wang [23, Conjecture 1.4] as well as Wigderson and Zuiddam [76, Section 13, p. 122]) states that the least possible exponent is shared by all concise tensors in $\mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d$:

▶ **Conjecture 2** (Extended asymptotic rank conjecture). *For all $d \geq 1$ it holds that $\sigma(d) = 1$.*

A key result supporting Conjecture 1 and Conjecture 2, implicit in Strassen [67, Proposition 3.6] and highlighted by Christandl, Vrana, and Zuiddam [17, Proposition 2.12] as well as Conner, Gesmundo, Landsberg, and Ventura [22, Remark 2.1], is that tensor rank is known to be nontrivially submultiplicative under Kronecker products; stated in terms of worst-case tensor exponents, Strassen proved that for all $d \geq 1$ it holds that $\sigma(d) \leq \frac{2\omega}{3}$.

Viewed in terms of exponents of tensors and universality, we can rephrase Strassen's result as stating that the exponent of the matrix multiplication tensor MM_2 controls from above the exponent of all other tensors, namely we have $\sigma(d) \leq \frac{4}{3}\sigma(\mathrm{MM}_2) = \frac{2\omega}{3}$. This rephrasing suggests that one should seek explicit constructions of tensors $U \in \mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d$ that are worst-case *universal* for the class with $\sigma(d) = \sigma(U)$. In rough analogy with computational complexity theory, such explicit tensors capture the "hardest" tensors in a class of tensors and, for example, would provide an explicit object of study towards resolving Conjectures 1 and 2.

1.2 Our Results – Explicit Universal Sequences of Zero-One-Valued Tensors

While we do not present explicit individual tensors U that are universal, as our main result we present an explicit *sequence* $\mathcal{U}$ of tensors that is universal and consists of zero-one-valued tensors in coordinates. Towards this end, let us extend the definition of the exponent of a tensor T to a sequence

$$\mathcal{T} = (T_j \in \mathbb{F}^{s_j} \otimes \mathbb{F}^{s_j} \otimes \mathbb{F}^{s_j} : j = 1, 2, \ldots) \tag{2}$$

of nonzero tensors. The *exponent* of the sequence $\mathcal{T}$ is the least nonnegative real $\sigma(\mathcal{T})$ such that $\mathrm{R}(T_j) \leq s_j^{\sigma(\mathcal{T})+o_j(1)}$. From (1) and (2) we immediately have $\sigma(T^{\boxtimes\mathbb{N}}) = \sigma(T)$ for all tensors $T \in \mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d$. Our main result is that there is an explicit sequence of tensors that characterizes the exponent of $\mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d$ and thus enables an approach towards resolving Strassen's conjecture.

▶ **Theorem 3** (Main; A universal sequence of tensors for d fixed). *For all $d \geq 1$ there is an explicit sequence $\mathcal{U}_d$ of zero-one-valued tensors with $\sigma(\mathcal{U}_d) = \sigma(d)$.*

In coordinates, the q^{th} tensor in the sequence $\mathcal{U}_d$ admits explicit combinatorial expression as a union of orbit-indicator tensors under a particular action of the symmetric group $\mathfrak{S}_q$. This combinatorial structure enables us to study the exponent $\sigma(\Delta)$ of the space of tensors spanned by $\{e_i \otimes e_j \otimes e_k : (i,j,k) \in \Delta\} \subseteq \mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d$ for a nonempty $\Delta \subseteq [d] \times [d] \times [d]$; when equality holds, we have $\sigma(\Delta) = \sigma(d)$.

▶ **Theorem 4** (Support-localized universal sequences of tensors). *For all nonempty $\Delta \subseteq [d] \times [d] \times [d]$ there is an explicit sequence $\mathcal{U}_\Delta$ of zero-one-valued tensors with $\sigma(\mathcal{U}_\Delta) = \sigma(\Delta)$.*

From Theorem 4 we obtain the following corollary for tight tensors and Strassen's conjecture (Conjecture 1). We need short preliminaries. We say that the set Δ is *tight* if there exist injective functions $\alpha, \beta, \gamma : [d] \to \mathbb{Z}$ such that $\alpha(i) + \beta(j) + \gamma(k) = 0$ for all $(i,j,k) \in \Delta$. A tensor $T \in \mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d$ is *tight* if it admits a coordinate-representation whose support is contained in a tight set.

▶ **Theorem 5** (A universal sequence of tensors for Strassen's conjecture). *For all $d \geq 1$ there is an explicit sequence $\mathcal{T}_d$ of tight zero-one-valued tensors with $\sigma(\mathcal{T}_d) = \sigma(\mathcal{T}_d)$.*

Further, by diagonalising the sequences $\mathcal{U}_d$ for increasing d we obtain a universal sequence providing worst possible exponent irrespective of d; that is, a universal sequence for the extended asymptotic rank conjecture (Conjecture 2):

▶ **Theorem 6** (A universal sequence of tensors for the extended asymptotic rank conjecture)**.** *There is an explicit sequence* $\mathcal{D} = (D_d : d = 1, 2, \ldots)$ *of zero-one-valued tensors with* $\lim_{d\to\infty} \sigma(D_d) = \lim_{d\to\infty} \sigma(d)$.

We note that the limit in the theorem above exists and is the supremum of all $\sigma(d)$; cf. Lemma 19. The above theorem may be regarded as a solution to the aforementioned hay in the haystack problem for the exponent of tensors.

1.3 Overview of Techniques

Before proceeding to review our further results, it will be convenient to give an overview of our main techniques and concepts underlying Theorem 3. In particular, a basis induced from integer compositions for the linear span of the image of the Kronecker power map $S \mapsto S^{\boxtimes q}$ will be our key tool.

The Kronecker power map $\mathbf{K_{d,q}}$ in coordinates. For a field $\mathbb{F}$ and positive integers d and q, our key object of study is the *Kronecker power map*

$$\mathrm{K}_{d,q} : \mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d \to \mathbb{F}^{d^q} \otimes \mathbb{F}^{d^q} \otimes \mathbb{F}^{d^q}$$

that takes a tensor $S \in \mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d$ to its q^{th} tensor (Kronecker) power $\mathrm{K}_{d,q}(S) = S^{\boxtimes q}$.

It will be convenient to study the Kronecker power map $\mathrm{K}_{d,q}$ by working with tensors in coordinates, so let us set up conventions accordingly. Let us write $[d] = \{1, 2, \ldots, d\}$ and identify the spaces $\mathbb{F}^{d\times d\times d} = \mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d$. For a tensor $S \in \mathbb{F}^{d\times d\times d}$ in the domain of $\mathrm{K}_{d,q}$, we write $S_{i,j,k} \in \mathbb{F}$ for the entry of S at position $i, j, k \in [d]$. For a tensor $T \in \mathbb{F}^{d^q\times d^q\times d^q}$ in the codomain of $\mathrm{K}_{d,q}$, it is convenient to index the entries $T_{I,J,K} \in \mathbb{F}$ of T with q-tuples $I = (i_1, i_2, \ldots, i_q) \in [d]^q$, $J = (j_1, j_2, \ldots, j_q) \in [d]^q$, $K = (k_1, k_2, \ldots, k_q) \in [d]^q$. Indeed, with this convention, the image $\mathrm{K}_{d,q}(S) = S^{\boxtimes q}$ of a tensor $S \in \mathbb{F}^{d\times d\times d}$ can be defined entrywise for all $I, J, K \in [d]^q$ by

$$S^{\boxtimes q}_{I,J,K} = S_{i_1,j_1,k_1} S_{i_2,j_2,k_2} \cdots S_{i_q,j_q,k_q} \,. \tag{3}$$

An explicit basis for the linear span of the image of $\mathbf{K_{d,q}}$. Next we embed the image $\mathrm{K}_{d,q}(\mathbb{F}^{d\times d\times d})$ in the least-dimensional subspace of $\mathbb{F}^{d^q\times d^q\times d^q}$ possible, so-called linear span, and provide an explicit basis for this subspace in the assumed coordinates. Towards this end, working with integer compositions that capture the distinct right-hand sides of (3) for a generic S with support in a nonempty set $\Delta \subseteq [d] \times [d] \times [d]$ will be convenient. In precise terms, a function $g : \Delta \to \mathbb{N}_{\geq 0}$ with $\sum_{\delta\in\Delta} g(\delta) = q$ is a *composition* of the positive integer q with domain Δ. Let us write $\mathcal{C}^{\Delta}_q$ for the set of all compositions of q with domain Δ. The distinct right-hand sides of (3) for a generic S are now enumerated by the compositions $g \in \mathcal{C}^{\Delta}_q$; explicitly, associate with g the product

$$S^g = \prod_{i,j,k\in[d]} S^{g(i,j,k)}_{i,j,k} \in \mathbb{F} \,. \tag{4}$$

Writing $\mathbb{F}^{\Delta}$ for the subspace of all tensors in $\mathbb{F}^{d\times d\times d}$ with support in Δ, we next provide a basis $(T^{(g)} \in \mathbb{F}^{d^q\times d^q\times d^q} : g \in \mathcal{C}^{\Delta}_q)$ that enables us to express the Kronecker power $S^{\boxtimes q}$ of an arbitrary tensor $S \in \mathbb{F}^{\Delta}$ as the linear combination

$$S^{\boxtimes q} = \sum_{g \in \mathcal{C}_q^\Delta} S^g T^{(g)} \,. \tag{5}$$

We need short preliminaries to define the tensors $T^{(g)}$ in the assumed coordinates. For $I, J, K \in [d]^q$, define the triple-counting composition $\Phi_{I,J,K} \in \mathcal{C}_q^{[d]\times[d]\times[d]}$ for all $u, v, w \in [d]$ by

$$\Phi_{I,J,K}(u,v,w) = |\{\ell \in [q] : i_\ell = u,\ j_\ell = v,\ k_\ell = w\}| \,. \tag{6}$$

We use *Iverson's bracket notation*; for a logical proposition P, we write $[\![P]\!]$ to indicate a 0 if P is false and a 1 if P is true.

▶ **Definition 7** (Composition basis for the linear span of the image $\mathrm{K}_{d,q}(\mathbb{F}^\Delta)$)**.** *For $g \in \mathcal{C}_q^\Delta$, define the tensor $T^{(g)} \in \mathbb{F}^{d^q \times d^q \times d^q}$ in coordinates for all $I, J, K \in [d]^q$ by*

$$\big(T^{(g)}\big)_{I,J,K} = [\![\Phi_{I,J,K} = g]\!] \,. \tag{7}$$

The tensors $(T^{(g)} : g \in \mathcal{C}_q^\Delta)$ are the composition basis *for the linear span of $\mathrm{K}_{d,q}(\mathbb{F}^\Delta)$.*

We provide various equivalent characterisations of these tensors $T^{(g)}$ in Lemma 9 (on page 10). The characterisation via integer compositions in Definition 7 in particular enables an immediate verification via (7), (4), and (3) that (5) holds. In particular, the linear span of the image $\mathrm{K}_{d,q}(\mathbb{F}^\Delta)$ is contained in the linear span of the composition basis. What is less immediate is that the reverse containment holds under mild assumptions on the field $\mathbb{F}$ by identifying the Kronecker power map with the Veronese map on homogeneous Δ-variate polynomials of degree q. In coordinates, by relying on homogeneous polynomial interpolation we show in Corollary 13 and Proposition 15 that for every $g \in \mathcal{C}_q^\Delta$ there exist tensors $S_f \in \mathbb{F}^\Delta$ and scalars $\lambda_{f,g} \in \mathbb{F}$ indexed by $f \in \mathcal{C}_q^\Delta$ such that

$$T^{(g)} = \sum_{f \in \mathcal{C}_q^\Delta} \lambda_{f,g} S_f^{\boxtimes q} \,. \tag{8}$$

Now (5) and (8) imply the composition basis spans exactly the linear span of $\mathrm{K}_{d,q}(\mathbb{F}^\Delta)$.

Sublinearity and serendipity of polynomial dimensionality of the span. By subadditivity of tensor rank, from the linear combination (5) we have immediately for an arbitrary tensor $S \in \mathbb{F}^\Delta$ that

$$\mathrm{R}\big(S^{\boxtimes q}\big) \le \sum_{g \in \mathcal{C}_q^\Delta} \mathrm{R}\big(T^{(g)}\big) \,.$$

Conversely, for an arbitrary $g \in \mathcal{C}_q^\Delta$, we have from (8) that

$$\mathrm{R}\big(T^{(g)}\big) \le \sum_{f \in \mathcal{C}_q^\Delta} \mathrm{R}\big(S_f^{\boxtimes q}\big) \,.$$

Recalling from combinatorics of integer compositions that $|\mathcal{C}_q^\Delta| = \binom{|\Delta|-1+q}{|\Delta|-1} \le (|\Delta| - 1 + q)^{|\Delta|-1}$, we observe that dimension of the linear span of $\mathrm{K}_{d,q}(\mathbb{F}^\Delta)$ grows only polynomially in q when Δ is fixed. Theorem 3 and Theorem 4 follow essentially immediately by taking $\mathcal{U}_\Delta = (U_{\Delta,q} : q = 1, 2, \ldots)$ with $U_{\Delta,q} = \bigoplus_{g \in \mathcal{C}_q^\Delta} T^{(g)}$ as the universal sequence.

Structure and description of the tensors $T^{(g)}$. The tensors $T^{(g)}$ admit several descriptions, e.g. a group-theoretic description can be given using orbit-indicators of the group $\mathfrak{S}_q$, see Lemma 9, and an alternative description can be given via type decompositions [76, Section 9.3] (see also [28]). There are several important applications of group theory in the study or tensor rank and asymptotic rank. We note that when G is an Abelian group then the structure tensor S_G of the group algebra is an orbit-indicator of the group G^2 identified with $\{(g_1, g_2, g_3) \in G^3 : g_1 + g_2 = g_3\}$; in this case, the tensor S_G has also minimal rank, equal to $|G|$, via the Discrete Fourier Transform. When G is not Abelian, the representation theory of G allows for nontrivial bounds on the rank of S_G; similar observations have been leveraged in the study of fast matrix multiplication; e.g. Cohn and Umans [20], Cohn, Kleinberg, Szegedy, and Umans [19], Cohn and Umans [21], Blasiak, Cohn, Grochow, Pratt, and Umans [10]. We expect this structure to enable further work towards an eventual proof or disproof of the (extended) asymptotic rank conjecture.

A win-win dichotomy. In addition to enabling worst-case characterisation of tensor exponents for families of tensors, the tensors $T^{(g)}$ enable the following "win-win" dichotomy (cf. Corollary 21 for a precise statement) that motivates their further study:

Either the extended asymptotic rank conjecture (Conjecture 2) holds, implying $\omega = 2$ and a disproof of the Set Cover Conjecture;

or the tensors $T^{(g)}$ form an explicit sequence of tensors with superlinear border rank growth, providing substantial progress for the "hay in the haystack" problem for rank and border rank.

A remark on explicitness. We stress that we here view explicitness as the property of not only a single tensor but rather a sequence of tensors; cf. e.g. [48, Section 3]. Both the tensors $T^{(g)}$ and the tensors in our universal sequences have zero-one entries in coordinates, and these entries may be computed fast. Indeed, from (7) it is immediate that we have linear-time algorithms that output $T^{(g)}_{I,J,K}$ given I, J, K, g as input. Similarly, listing (or ranking/unranking) integer compositions $g \in \mathcal{C}_q^\Delta$ for given Δ, q admit fast algorithms.

Invariant Specht tensors. Our techniques reduce the question about asymptotic ranks to questions about ranks of specific tensors, where $T^{(g)}$ is only one possible choice. Another one is to apply representation theory of the symmetric group $\mathfrak{S}_p$. It turns out that ranks of very special tensors, precisely invariants in the tensor product of three Specht modules $(S^\alpha \otimes S^\beta \otimes S^\gamma)^{\mathfrak{S}_p}$, where α, β, γ are partitions of p with at most d parts, govern the exponent $\sigma(d)$. Precise results are given in Section 4.

1.4 Further Results

Our second result relates extended Strassen's conjecture to equations of varieties, in particular secant varieties. We tacitly assume sufficient background in algebraic geometry and geometry of tensors (e.g. [27, 42, 55]). Also, unless otherwise mentioned, we assume that the field $\mathbb{F}$ is the field of complex numbers.

To set the context, it is a well-established method to prove that tensors have high border rank by exhibiting polynomials vanishing on the k^{th} secant variety of the Segre variety $X = (\mathbb{P}^{d-1})^{\times 3}$ and evaluating it on a given tensor T. Among the state of the art theoretical equations in this context are the Koszul flattenings and the Young flattenings, which can provide border rank bounds up $(2 - \epsilon)d$ for any $\epsilon > 0$ for $d \gg 0$. The degree of those equations grows as a polynomial in d for fixed ϵ, however the degree of this polynomial also

grows as $\epsilon \to 0$. The second method to obtain equations of secant varieties is computational, based on representation theory and linear algebra. This is an exhaustive method, finding all equations in the given degree and partitioning them into so-called isotypic components [11, 36]. Still of course its scope is limited due to computational obstacles. For border rank $k < d$, the smallest degree of a polynomial vanishing on the k^{th} secant variety is exactly $k + 1$ [46]. However, for border rank above d, one observes fast grow of the minimal degree in which equations exist; for example, the smallest degree of an equation vanishing on the 6^{th} secant variety of $(\mathbb{P}^3)^{\times 3}$ is 19 and on the 18^{th} secant variety of $(\mathbb{P}^6)^{\times 3}$ is at least 187000 [36]. The lack of low-degree (or otherwise easy) equations has been perceived so far as an obstacle in proving that tensors have high border rank, in particular in disproving Strassen's conjecture or its extensions. In this paper, we proceed in the *opposite* direction, namely that *absence* of low-degree equations of secant varieties implies upper bound on $\sigma(d)$.

▶ **Theorem 8** (Absence of low-degree equations implies low asymptotic rank)**.** *Let $X \subseteq \mathbb{P}((\mathbb{F}^d)^{\otimes 3})$ be a variety contained in the locus of tensors of asymptotic rank at most r. Suppose that no polynomial of degree p vanishes on X. Then every tensor in $(\mathbb{F}^d)^{\otimes 3}$ has asymptotic rank at most*

$$r\binom{d^3 - 1 + p}{d^3 - 1}^{\frac{1}{p}}.$$

We note that the theorem above implies that bounds on asymptotic rank of special tensors may imply bounds for all tensors. As the variety X may be always assumed to be $GL(d)^{\times 3}$ invariant one may combine the computational method of finding isotypic decomposition of homogeneous polynomials with obtaining good bounds on p, in order to obtain new bounds on $\sigma(d)$. We leave this line of research for the future.

1.5 Related Work

It is known that computing the tensor rank of a given tensor is NP-hard [35]; see also [37, 60, 63] for pertinent hardness results. It is also difficult in practice to determine the rank and border rank of small tensors; for example, the rank, border rank and border support rank of MM_2 are known to be seven [9, 36, 41, 70].

There has been extensive interest in constructing explicit tensors of high rank or border rank [1, 44, 48, 51]. This study is motivated by the need for new methods to provide lower complexity bounds. Currently, we do not know how to construct sequences of explicit tensors $T \in \mathbb{F}^d \otimes \mathbb{F}^d \otimes \mathbb{F}^d$ of rank or border rank above $3d$. In fact, there are no known examples of tensors with entries 0 or 1 and rank or border rank greater than $3d$. The only method to construct such tensors, is by making the entries incomparable in size, i.e. each entry is of different order of magnitude then other entries, or making them algebraically independent, which makes the tensors very far from explicit. However, it is easy to prove that tensors of super-linear border rank, with respect to their size exist. Precisely, for any constant $C > 0$ and d sufficiently large there exists a tensor $T \in (\mathbb{F}^d)^{\otimes 3}$ of border rank greater than Cd. Even more: general tensors will have greater border rank than Cd for d sufficiently large and the growth of maximal border rank is quadratic in d. We simply do not know how to provide explicit examples of such tensors, as the methods we have do not allow to prove that particular tensors have large rank. These obstructions are related to the fact that the cactus variety, that contains the secant variety, fills the whole ambient space, while most of the equations we know for secant varieties, also vanish on cactus varieties [12, 33, 32, 38, 49, 73]. We even do not know how to provide a sequence of explicit tensors so that infinitely many elements of the sequence would have high rank, say above $3d$.

The families of tensors with fixed support are also studied. An important concept is that of support rank [21], which has given rise to other support-based algorithm-design techniques (e.g. [3, 39]). Further, bounding asymptotic rank of special tensors was recently tied to NP-hard problems; in particular, Strassen's conjecture (Conjecture 1) would imply unexpected (but not polynomial) upper bounds on complexity of (randomized) algorithms for NP-hard problems [7, 59].

Beyond the present invariant Specht tensors, representation theory of the symmetric and general linear groups has extensive connections to algebraic complexity theory via Strassen's theory of asymptotic spectra (e.g. [18]) as well as the geometric complexity theory program (e.g. [56, 57]).

1.6 Organization of This Paper

Section 2 reviews notational preliminaries and definitions. Section 3 studies the composition basis and proves our main theorems on explicit universal sequences of tensors (Theorems 3 to 6). Section 4 introduces invariant Specht tensors and shows their universality for the exponent $\sigma(d)$ (Corollary 38). Section 5 relates the equations of varieties to bounds on asymptotic rank and proves Theorem 8.

2 Preliminaries

Recall that we write $[d] = \{1, 2, \ldots, d\}$. The set $[d]^q$ consists of sequences of length q of integers in $[d]$. We fix the canonical basis $e_1, e_2, \ldots, e_m \in \mathbb{F}^m$.

In this article, we exclusively work with tensors of format $a \times b \times c$, where in most cases $a = b = c$. These are elements of the vector space $\mathbb{F}^a \otimes \mathbb{F}^b \otimes \mathbb{F}^c \simeq \mathbb{F}^{a\times b\times c}$. In analogy to the case of matrices, the reader may freely think about tensors as three dimensional arrays filled with elements of $\mathbb{F}$. For a vector $v \in \mathbb{F}^a$ we write $v_i := e_i^*(v) \in \mathbb{F}$ for $1 \leq i \leq a$. We write $S_{i,j,k} \in \mathbb{F}$ for the entry of $S \in \mathbb{F}^a \otimes \mathbb{F}^b \otimes \mathbb{F}^c$ at position $(i, j, k) \in [a] \times [b] \times [c]$.

For three vectors $v_1 \in \mathbb{F}^a, v_2 \in \mathbb{F}^b$ and $v_3 \in \mathbb{F}^c$ we define the tensor $v_1 \otimes v_2 \otimes v_3 \in \mathbb{F}^a \otimes \mathbb{F}^b \otimes \mathbb{F}^c$ where the (i, j, k) coordinate equals $(v_1)_i(v_2)_j(v_3)_k$. Tensors of this form are called *rank one* tensors. The *rank* of a tensor T is the smallest r such that T is sum of r rank one tensors.

In case $\mathbb{F} = \mathbb{R}$ or $\mathbb{F} = \mathbb{C}$ we define the *border rank* of a tensor T as the smallest r such that in any neighbourhood of T there exists a tensor of rank r. For more details about rank and border rank we refer to [16, 42, 55]. The Kronecker power of a tensor is defined as in formula (3). *Asymptotic rank* [34] of a tensor T is defined as $\lim_{n\to\infty} \mathrm{R}(T^{\boxtimes n})^{\frac{1}{n}}$. We write $\langle s\rangle$ to be the unit tensor $\langle s\rangle = \sum_{i=1}^{s} e_i \otimes e_i \otimes e_i$.

3 Universal Sequences of Tensors for the Asymptotic Rank Conjecture

In this section we prove that the tensors $T^{(g)}$ in the composition basis (Definition 7) form a universal family for the asymptotic rank conjecture. We start with a more detailed analysis of the structure of the composition basis for fixed d (Section 3.1), and follow with an analysis for increasing d (Section 3.2); in particular, we establish the existence of the limit exponent $\lim_{d\to\infty} \sigma(d)$. Theorem 3 and Theorem 6 are restated and proved next (Section 3.3). Using techniques of Strassen, we then show that the universal sequences have nontrivially low tensor rank (Section 3.4). We end this section by setting up our techniques for support-localization (Section 3.5) as well as restate and prove Theorem 5, our main result for tight tensors (Section 3.6).

3.1 Properties of the Composition Basis

We start with the following easy lemma that provides various alternative characterisations of tensors $T^{(g)}$ in Definition 7.

We note that the symmetric group $\mathfrak{S}_q$ acts by permutations on $\mathbb{F}^{[d]^q}$ and diagonally on $(\mathbb{F}^{[d]^q})^{\otimes 3}$.

▶ **Lemma 9** (Equivalent definitions of the composition basis). *We have the following equivalent definitions of tensors $T^{(g)}$ for any $g \in \mathcal{C}_q^{[d]^3}$.*

1. *The tensors $T^{(g)}$ are precisely sums of $\mathfrak{S}_q$ orbits of canonical basis tensors in $(\mathbb{F}^{[d]^q})^{\otimes 3}$.*
2. *The tensors $T^{(g)}$ are coefficients of monomials for the Kronecker map $K_{d,q}$, explicitly:*

$$K_{d,q}(S) = \sum_{g \in \mathcal{C}_q^{[d]^3}} S^g T^{(g)} .$$

   In particular, the linear span of the image is contained in the linear span of tensors $T^{(g)}$.
3. *If $|\mathbb{F}| > q$, then the tensors $T^{(g)}$, up to scaling by a constant, are precisely tensors with inclusion minimal supports in the linear span of the image of $K_{d,q}$.*

Proof. We note that the condition $\Phi_{I,J,K} = g$ from Definition 7 determines the triple (I, J, K) up to simultaneous action by an element of $\mathfrak{S}_q$. This proves equivalence of point (1) with the original definition.

The coordinates of $K_{d,p}(S)$ in the canonical basis are monomials of degree q in the coordinates of S. A monomial S^g appears exactly on the entries indexed by such (I, J, K)'s that $\Phi_{I,J,K} = g$. This proves equivalence of point (2) with Definition 7.

In Proposition 15, we show that the linear span of the image of $K_{d,q}$ coincides with the linear span of $T^{(g)}$'s for $g \in \mathcal{C}_q^{[d]^3}$ and $|\mathbb{F}| > q$. As $T^{(g)}$'s have disjoint supports, we obtain point (3). ◀

▶ Remark 10 (Invariant subspaces defined by marginals of g). It is immediate that $T^{(g)} \in (\mathbb{F}^{[d]^q})^{\otimes 3}$, however the tensor $T^{(g)}$ also belongs to smaller invariant subspace defined by g. Namely, for $g \in \mathcal{C}_q^{[d]^3}$ let $g_1 : [d] \to \mathbb{N}$ be the first marginal of g; that is, let $g_1(j) = \sum_{a,b \in [d]} g(j, a, b)$ for all $j \in [d]$. In the same way, define the second marginal g_2 and third marginal g_3. For $i = 1, 2, 3$, let $U_i \subseteq [d]^q$ be the set of all q-tuples $I \in [d]^q$ such that the value j appears in I exactly $g_i(j)$ times for all $j \in [d]$. We have $|U_i| = \binom{q}{g_i(1), g_i(2), \ldots, g_i(d)}$ as well as $T^{(g)} \in \mathbb{F}^{U_1} \otimes \mathbb{F}^{U_2} \otimes \mathbb{F}^{U_3}$. Clearly, the ambient space is $\mathfrak{S}_q$ invariant.

▶ **Example 11** (The small Coppersmith–Winograd tensor). Let $d = 2$, $q = 3$ and g assign value 1 on $(0, 0, 1)$, $(0, 1, 0)$ and $(1, 0, 0)$. The tensor $T^{(g)}$ is the small Coppersmith–Winograd tensor in $(\mathbb{F}^3)^{\otimes 3}$, that is:

$$e_0 \otimes e_1 \otimes e_2 + e_0 \otimes e_2 \otimes e_1 + e_1 \otimes e_0 \otimes e_2 + e_1 \otimes e_2 \otimes e_0 + e_2 \otimes e_0 \otimes e_1 + e_2 \otimes e_1 \otimes e_0 .$$

▶ **Definition 12** (Linear span of the composition basis). *Let $L_{d,q}$ be the linear span of the tensors $T^{(g)}$ for $g \in \mathcal{C}_q^{[d]^3}$. As the tensors $T^{(g)}$ have disjoint supports they are linearly independent and hence form a basis of $L_{d,q}$. We note that $\dim L_{d,q} = \binom{d^3+q-1}{q}$ which is the cardinality of $\mathcal{C}_q^{[d]^3}$.*

▶ **Corollary 13** (Span coincides with the space of invariants). *The tensors $T^{(g)}$ for $g \in \mathcal{C}_q^{[d]^3}$ form a basis of the invariants space $((\mathbb{F}^{[d]^q})^{\otimes 3})^{\mathfrak{S}_q}$, which coincides with $L_{d,q}$. Up to rescaling, it is the unique basis of that space made of tensors with disjoint supports.*

From now on we will assume that the cardinality of the field is greater than q.

▶ **Lemma 14** (Dual space of homogeneous polynomials). *There is an isomorphism of vector spaces: $L^*_{d,q}$ of linear forms and functions on $(\mathbb{F}^d)^{\otimes 3}$ given by homogeneous polynomials of degree q. The isomorphism sends a linear form l to the polynomial function $l \circ K_{d,q}$.*

Proof. Let $T^{(g)*}$ be the basis of $L^*_{d,q}$ dual to $T^{(g)}$. The image of the linear function $\sum \lambda_g T^{(g)*}$ is $\sum \lambda_g S^g$. Surjectivity is obvious.

By induction on the number n of variables, one proves that no polynomial of degree q may vanish identically on $\mathbb{F}^n$, when $|\mathbb{F}| > q$. Hence, no linear form l is mapped to the zero function and the linear map $l \mapsto l \circ K_{d,p}$ is injective. This finishes the proof. ◀

▶ **Proposition 15** (Span of the image of the Kronecker power map). *The space $L_{d,q}$ is the linear span of the image of $K_{d,q}$.*

Proof. Clearly $L_{d,q}$ contains the image of $K_{d,q}$. If the containment was strict, there would exist a nonzero linear function $l \in L_{d,q}$ vanishing on the image. Then $l \circ K_{d,q} = 0$. This is not possible by Lemma 14. ◀

▶ Remark 16 (The Veronese map of degree q). Up to isomorphism, $K_{d,q}$ may be identified with the q^{th} Veronese map, that is a map defined by all degree q monomials. However, in coordinates each monomial may appear more than once, as each monomial represented by $g \in \mathcal{C}_q^{[d]^3}$ appears on the support of $T^{(g)}$.

▶ **Lemma 17** (Maximum rank in the composition basis controls rank in $L_{d,q}$). *Let r be the maximum rank (respectively, asymptotic rank) of $T^{(g)}$ over $g \in \mathcal{C}_q^{[d]^3}$. Every tensor in $L_{d,q}$ has rank (respectively, asymptotic rank) at most*

$$r|\mathcal{C}_q^{[d]^3}| = r\binom{d^3-1+q}{d^3-1}.$$

In particular, every tensor in $(\mathbb{F}^d)^{\otimes 3}$ has asymptotic rank at most

$$\left(r\binom{d^3-1+q}{d^3-1}\right)^{\frac{1}{q}}.$$

Proof. Fix $T \in (\mathbb{F}^d)^{\otimes 3}$. By Lemma 9 the tensor $K_{d,q}(T)$ is a linear combination of $T^{(g)}$'s for $g \in \mathcal{C}_q^{[d]^3}$. As $|\mathcal{C}_q^{[d]^3}| = \binom{d^3-1+q}{d^3-1}$ we obtain:

$$\mathrm{R}(K_{d,q}(T)) \leq r\binom{d^3-1+q}{d^3-1}.$$

The statement under the assumption of asymptotic rank r for $T^{(g)}$'s is proved in the same way, noting that asymptotic rank is subadditive and submultiplicative (e.g. [76]). ◀

3.2 The Extended Asymptotic Rank Conjecture and the Limit Exponent

We are now ready to connect the composition-basis tensors $T^{(g)}$ to the extended asymptotic rank conjecture (Conjecture 2).

▶ **Corollary 18** (The composition basis suffices for the extended asymptotic rank conjecture). *If there exists an infinite set $S = \{d_1, d_2, \ldots\} \subseteq \mathbb{Z}_{\geq 1}$ such that for any d_i there exist infinitely many $q_{i,j} \in \mathbb{Z}_{\geq 1}$ such that Conjecture 2 holds for the tensor $T^{(g)}$ for all $g \in \mathcal{C}_{q_{i,j}}^{[d_i]^3}$, then Conjecture 2 holds for all tensors.*

Proof. First fix d_i. By Lemma 17 we see that any $T \in ((\mathbb{F}^{d_i})^{\otimes 3})$ has asymptotic rank at most:

$$(d_i^q)^{\frac{1}{q}} \binom{d_i^3 - 1 + q}{d_i^3 - 1}^{\frac{1}{q}}.$$

Note that for fixed d_i we have

$$\lim_{q\to\infty} \binom{d_i^3 - 1 + q}{d_i^3 - 1}^{\frac{1}{q}} = 1\,.$$

This confirms Conjecture 2 for any $T \in (\mathbb{F}^{d_i})^{\otimes 3}$. We conclude that it must hold for each d from the next Lemma 19. ◀

▶ **Lemma 19** (Existence of the limit exponent). *The limit* $\lim_{d\to\infty} \sigma(d)$ *exists and equals the supremum of the set* $\{\sigma(d) : d \in \mathbb{Z}_{\geq 1}\}$.

Proof. As the sequence $\sigma(d)$ is bounded it is enough to prove:

$$\forall_{d_0\in\mathbb{Z}_{\geq 1}} \forall_{\epsilon>0} \exists_{D\in\mathbb{Z}_{\geq 1}} \forall_{d>D} \quad \sigma(d) > \sigma(d_0) - \epsilon\,.$$

Fix $\epsilon > 0$ and $d_0 \in \mathbb{Z}_{\geq 1}$. For contradiction assume there are infinitely many d_i such that $\sigma(d_i) \leq \sigma(d_0) - \epsilon$. Hence, we also obtain infinitely many k_i such that $d_0^{k_i} \leq d_i < d_0^{k_i+1}$.

We note that we have $\sigma(d_0^{k_i}) \geq \sigma(d_0)$ as taking Kronecker power of a tensor does not change the exponent. Hence, for any $\delta > 0$ there exists $T_\delta \in (\mathbb{F}^{d_0^{k_i}})^{\otimes 3}$ of asymptotic rank at least $d_0^{k_i(\sigma(d_0)-\delta)}$. As we may embed T_δ in larger space, we see that its asymptotic rank is at most $d_i^{\sigma(d_i)} \leq d_0^{(k_i+1)(\sigma(d_0)-\epsilon)}$. We obtain:

$$k_i(\sigma(d_0) - \delta) \leq (k_i + 1)(\sigma(d_0) - \epsilon).$$

As this holds for any $\delta > 0$ we obtain:

$$k_i\sigma(d_0) \leq (k_i + 1)(\sigma(d_0) - \epsilon),$$

which is a contradiction for k_i sufficiently large. ◀

The following more precise result shows that the exponent governing the asymptotic rank is completely determined via asymptotic ranks of the tensors $T^{(g)}$.

▶ **Corollary 20** (The composition bases govern tensor exponents). *For all* $\tau \geq 0$ *it holds that each tensor* $T^{(g)} \in (\mathbb{F}^m)^{\otimes 3}$ *(where* m *is uniquely determined by* g*) has asymptotic rank at most* m^τ *if and only if any tensor* $T \in (\mathbb{F}^d)^{\otimes 3}$ *(where* d *is arbitrary) has asymptotic rank at most* d^τ*. In particular, for any tensor* T *we have* $\sigma(T) \leq \sup_g \sigma(T^{(g)})$.

Proof. The implication "⇐" is immediate. For the implication "⇒", assume that a tensor T has asymptotic rank greater than $d^{\tau+\epsilon}$ for fixed $\epsilon > 0$. By applying Lemma 17 for $q \gg 0$ such that $\binom{d^3-1+q}{d^3-1}^{\frac{1}{q}} < d^\epsilon$ we obtain a contradiction. ◀

We note that there are countably many tensors $T^{(g)}$ and they form a sequence of explicit tensors. This sequence gives rise to the following dichotomy.

▶ **Corollary 21** (A dichotomy on improved algorithms or explicit rank lower bounds). *We have the following dichotomy. Either*

- **(i)** *the extended asymptotic rank conjecture (Conjecture 2) holds; that is, for all tensors T we have $\sigma(T) \leq 1$, which implies $\omega = 2$ and disproves the Set Cover Conjecture; or*
- **(ii)** *the tensors $T^{(g)}$ form a sequence of explicit tensors such that for any constant $C \in \mathbb{Z}_{\geq 1}$ there exist infinitely many $T^{(g)} \in (\mathbb{F}^m)^{\otimes 3}$, such that rank, border rank, and asymptotic rank is greater than Cm.*

It is clear that for special g the rank of $T^{(g)}$ can be small. For example if g is supported at one element, then $T^{(g)}$ has support of cardinality one and thus is of rank one.

3.3 Explicit Universal Sequences and Main Results

We first prove our main theorem for fixed d, and define the universal sequence accordingly.

▶ **Definition 22** (Universal sequence for a fixed d). *For a fixed positive integer d, we define the sequence $\mathcal{U}_d = (U_{d,q} : q = 1, 2, \ldots)$ of tensors, where we set*

$$U_{d,q} = \bigoplus_{g \in \mathcal{C}_q^{[d]^3}} T^{(g)}.$$

We are now ready to prove our main theorem; Theorem 3 is restated below for convenience.

▶ **Theorem 23** (Main; Explicit universal sequences of tensors). *For all $d \geq 1$ we have $\sigma(\mathcal{U}_d) = \sigma(d)$.*

Proof. We prove the equality by proving both inequalities.

First, suppose that $\sigma(\mathcal{U}_d) < \sigma$ for some constant σ. Fix an $\epsilon > 0$ so that for q sufficiently large $\mathrm{R}(U_{d,q}) < (|\mathcal{C}_q^{[d]^3}| d^q)^{\sigma-\epsilon}$. Then, for q sufficiently large and all $g \in \mathcal{C}_q^{[d]^3}$ we have $\mathrm{R}(T^{(g)}) \leq \mathrm{R}(U_{d,q}) < (|\mathcal{C}_q^{[d]^3}| d^q)^{\sigma-\epsilon}$. As $\lim_{q\to\infty} |\mathcal{C}_q^{[d]^3}|^{1/q} = 1$ we see that for q sufficiently large and all $g \in \mathcal{C}_q^{[d]^3}$ we have $\sigma(T^{(g)}) \leq \sigma$. Applying Lemma 17 and taking limit, as $q \to \infty$ we see that $\sigma(T) \leq \sigma$ for all $T \in (\mathbb{F}^d)^{\otimes 3}$.

Suppose now that for all $T \in (\mathbb{F}^d)^{\otimes 3}$ we have $\sigma(T) < \sigma$. Fix $\epsilon > 0$, such that $\sigma(T) < \sigma - \epsilon$ for all $T \in (\mathbb{F}^d)^{\otimes 3}$. By Proposition 15 for any $g \in \mathcal{C}_q^{[d]^3}$ the tensor $T^{(g)}$ is a sum of at most $|\mathcal{C}_q^{[d]^3}|$-many tensors of type $K_{d,q}(T)$. Hence, for q sufficiently large we have:

$$\mathrm{R}(T^{(g)}) \leq |\mathcal{C}_q^{[d]^3}| d^{q(\sigma-\epsilon)}.$$

It follows that:

$$\mathrm{R}(U_{d,q}) \leq |\mathcal{C}_q^{[d]^3}|^2 d^{q(\sigma-\epsilon)}.$$

As $\lim_{q\to\infty} |\mathcal{C}_q^{[d]^3}|/d^{\epsilon q} = 0$ we see that $\mathrm{R}(U_{d,q}) \leq (|\mathcal{C}_q^{[d]^3}| d^q)^{\sigma}$ for $q \gg 0$. Thus $\sigma(\mathcal{U}_d) \leq \sigma$, which finishes the proof. ◀

Next, we construct one universal sequence realizing the worst possible exponent, irrespective of d via a diagonal argument on the sequences $\mathcal{U}_d$. Accordingly, recalling Definition 22, define the diagonal sequence $\mathcal{D} = (D_d : d = 1, 2, \ldots)$ of tensors for all positive integers d by setting $D_d = U_{d,d^4}$. It is immediate that the sequence is explicit. Theorem 6 is restated below for convenience.

▶ **Theorem 24** (An explicit universal sequence for the extended asymptotic rank conjecture). *The sequence $\mathcal{D}$ satisfies $\lim_{d\to\infty} \sigma(D_d) = \lim_{d\to\infty} \sigma(d)$.*

Proof. By Lemma 19 it is enough to prove that for every $\epsilon > 0$, for all d sufficiently large we have $\sigma(U_{d,d^4}) > \sigma(d) - \epsilon$. For any (concise) $T \in (\mathbb{F}^d)^{\otimes 3}$ we have:

$$d^{d^4\sigma(T)} = \tilde{\mathrm{R}}(T)^{d^4} \leq \sum_{g \in \mathcal{C}_{d^4}^{[d]^3}} \tilde{\mathrm{R}}(T^{(g)}) \leq |\mathcal{C}_{d^4}^{[d]^3}| \tilde{\mathrm{R}}(U_{d,d^4}) \leq |\mathcal{C}_{d^4}^{[d]^3}| (|\mathcal{C}_{d^4}^{[d]^3}| d^{d^4})^{\sigma(U_{d,d^4})}.$$

After taking root of order d^4 we only need to prove that $\lim_{d\to\infty} |\mathcal{C}_{d^4}^{[d]^3}|^{1/d^4} = 1$. We have:

$$\lim_{d\to\infty} |\mathcal{C}_{d^4}^{[d]^3}|^{1/d^4} = \lim_{d\to\infty} \binom{d^4 + d^3 - 1}{d^3 - 1}^{1/d^4} \leq \lim_{d\to\infty} (2d^4)^{d^3/d^4} = 1,$$

which finishes the proof. ◀

3.4 Upper Bounds on Tensor Rank in the Universal Sequences

A natural question arises: what are ranks of the tensors $T^{(g)}$. In order to provide upper bounds, we recall the following lemma due to Strassen (implicit in [67, Proposition 3.6]).

▶ **Lemma 25** (Limited universality of matrix multiplication tensors)**.** *The matrix multiplication tensor* MM_{d^2} *degenerates to* $T^{\boxtimes 3}$ *for any tensor* $T \in (\mathbb{F}^d)^{\otimes 3}$*. In particular,* $\sigma(T) \leq 2\omega/3$*.*

Proof. Let us write in coordinates $T = (\lambda_{p,q,r})$ and

$$\mathrm{MM}_{d^2} = \sum_{i_1,i_2,j_1,j_2,k_1,k_2 \in [d]} e_{j_1,j_2}^{i_1,i_2} \otimes e_{k_1,k_2}^{j_1,j_2} \otimes e_{i_1,i_2}^{k_1,k_2}.$$

For $i = 1, 2, 3$, define $l_i : \mathbb{F}^{d^4} \to \mathbb{F}^{d^3}$ as follows:

$$l_1(e_{j_1,j_2}^{i_1,i_2}) = \sum_{a \in [d]} \lambda_{a,j_1,i_1} e_{a,j_2,i_2},$$
$$l_2(e_{k_1,k_2}^{j_1,j_2}) = \sum_{b \in [d]} \lambda_{j_2,b,k_1} e_{j_1,b,k_2},$$
$$l_3(e_{i_1,i_2}^{k_1,k_2}) = \sum_{c \in [d]} \lambda_{i_2,k_2,c} e_{i_1,k_1,c}.$$

We have $l_1 \otimes l_2 \otimes l_3(\mathrm{MM}_{d^2}) = T^{\boxtimes 3}$. ◀

In a similar way we obtain:

▶ **Lemma 26** (Limited universality of matrix multiplication under powers)**.** *The matrix multiplication tensor* $\mathrm{MM}_{d^{2k}}$ *degenerates to* $T^{\boxtimes 3k}$ *for any tensor* $T \in (\mathbb{F}^d)^{\otimes 3}$*.*

We obtain the following corollary, which currently is asymptotically in $k \gg 0$ the best one we know, that works for arbitrary $g \in \mathcal{C}_{3k}^{[d]^3}$.

▶ **Corollary 27** (Upper bound on rank via matrix multiplication)**.** *For any* $g \in \mathcal{C}_{3k}^{[d]^3}$ *the tensor* $T^{(g)}$ *has rank at most* $\mathrm{R}(\mathrm{MM}_{d^{2k}}) \cdot \binom{d^3-1+3k}{d^3-1}$.

▶ Remark 28 (Note on special g). We note that Corollary 27 may be improved for many special g. For any $g \in \mathcal{C}_{3k}^{[d]^3}$ we have three marginal distributions on $[d]$. If such a distribution is close to uniform, then we obtain about d^k compatible sequences of length k; that is, asymptotically every sequence is compatible. However, if the distribution is far from uniform, we obtain less sequences, say d^{ck} for some $c < 1$. In such a case $\mathrm{MM}_{d^{2k}}$ in Lemma 26 may be changed into smaller matrix multiplication.

3.5 Support-Localized Universality

Many of our results remain true if we consider tensors $T \in (\mathbb{F}^d)^{\otimes 3}$ with support contained in a nonempty $\Delta \subseteq [d]^3$. Let us write $\mathbb{F}^\Delta$ be the linear space of tensors with such a support. As before $K_{d,q}$ restricted to $\mathbb{F}^\Delta$ is still a Veronese map, simply in variables corresponding to elements from Δ. The image of $K_{d,q}|_{\mathbb{F}^\Delta}$ has a basis made of tensors $T^{(g)}$ for $g \in \mathcal{C}_q^\Delta$. Below we present a variant of Lemma 17, which proof is analogous.

▶ **Lemma 29** (Maximum rank in the support-localized composition basis controls rank in $\mathbb{F}^\Delta$)**.** *Let r be the maximum rank (respectively, asymptotic rank) of $T^{(g)}$ over $g \in \mathcal{C}_q^\Delta$. The q^{th} Kronecker power of every tensor in $\mathbb{F}^\Delta$ has rank (respectively, asymptotic rank) at most*

$$r|\mathcal{C}_q^\Delta| = r\binom{|\Delta|-1+q}{|\Delta|-1}.$$

In particular, every tensor in $(\mathbb{F}^d)^{\otimes 3}$ with support contained in Δ has asymptotic rank at most

$$\left(r\binom{|\Delta|-1+q}{|\Delta|-1}\right)^{\frac{1}{q}}.$$

Next, we present a variant of Strassen's result for tensors with fixed support. For $\Delta \subseteq [d]^3$, we will use the following notation: $\Delta_{1,2} = \{(i,j) : \exists_k (i,j,k) \in \Delta\}$, $\Delta_{1,3} = \{(i,k) : \exists_j (i,j,k) \in \Delta\}$, and $\Delta_{2,3} = \{(j,k) : \exists_i (i,j,k) \in \Delta\}$.

Consider the following support-localized restriction of the matrix multiplication tensor MM_{d^2}:

$$\mathrm{MM}_\Delta := \sum_{\substack{(j_1,i_1)\in\Delta_{2,3}\\(j_2,k_1)\in\Delta_{1,3}\\(i_2,k_2)\in\Delta_{1,2}}} e_{j_1,j_2}^{i_1,i_2} \otimes e_{k_1,k_2}^{j_1,j_2} \otimes e_{i_1,i_2}^{k_1,k_2}.$$

We obtain the following version of Lemma 25.

▶ **Lemma 30** (Limited universality of support-localized matrix multiplication)**.** *The tensor MM_Δ degenerates to $T^{\boxtimes 3}$ for any tensor $T \in (\mathbb{F}^d)^{\otimes 3}$ with support contained in Δ.*

▶ **Corollary 31** (Upper bound on rank via support-localized matrix multiplication)**.** *For any $g \in \mathcal{C}_{3k}^\Delta$ the tensor $T^{(g)}$ has rank at most $\mathrm{R}(\mathrm{MM}_\Delta) \cdot \binom{|\Delta|-1+3k}{|\Delta|-1}$.*

Next we present the proof of Theorem 4.

▶ **Theorem 32** (Support-localized universal sequences of tensors)**.** *For all nonempty $\Delta \subseteq [d] \times [d] \times [d]$ there is an explicit sequence $\mathcal{U}_\Delta$ of zero-one-valued tensors with $\sigma(\mathcal{U}_\Delta) = \sigma(\Delta)$.*

Proof. Let $\mathcal{U}_\Delta = (U_{\Delta,q} : q = 1, 2, \dots)$ for

$$U_{\Delta,q} = \bigoplus_{g\in\mathcal{C}_q^\Delta} T^{(g)}.$$

As the image of $K_{d,q}|_{\mathbb{F}^\Delta}$ has a basis made of tensors $T^{(g)}$ for $g \in \mathcal{C}_q^\Delta$ the proof is analogous to the proof of Theorem 23 where we replace reference to Lemma 17 by reference to Lemma 29. ◀

3.6 Tight Tensors and Strassen's Conjecture

We now proceed to our main theorem for tight tensors and fixed d; we start by defining a corresponding universal sequence.

▶ **Definition 33** (Universal sequence for tight tensors and fixed d). *For a fixed positive integer d, we define the sequence $\mathcal{T}_d = (T_{d,q} : q = 1, 2, \ldots)$ of tensors, where we set*

$$T_{d,q} = \bigoplus_{\substack{\Delta \subseteq [d]^3 \\ \Delta \text{ tight}}} U_{\Delta,q} = \bigoplus_{\substack{\Delta \subseteq [d]^3 \\ \Delta \text{ tight}}} \bigoplus_{g \in \mathcal{C}_q^\Delta} T^{(g)} .$$

Next we prove Theorem 5, which we restate below for convenience.

▶ **Theorem 34** (A universal sequence of tensors for Strassen's conjecture). *We have $\sigma(\mathcal{T}_d) = \sigma(\mathcal{T}_d)$ and each $T_{d,q}$ is tight.*

Proof. First, we prove that for $g \in \mathcal{C}_q^\Delta$, where Δ is tight, the tensor $T^{(g)}$ is tight. Indeed, if we take any tensor T with support Δ, it is tight, thus so is $T^{\boxtimes q}$. As $T^{(g)}$ has smaller support it is tight. As direct sum of tight tensors is tight, we see that each $T_{d,q}$ is tight.

Next we note that for fixed d each $T_{d,q}$ is a sum of a fixed number of tensors $U_{\Delta,q}$. Thus, there exists a constant C such that:

$$\max_{\substack{\Delta \subseteq [d]^3 \\ \Delta \text{ tight}}} \mathrm{R}(U_{\Delta,q}) \leq \mathrm{R}(T_{d,q}) \leq C \max_{\substack{\Delta \subseteq [d]^3 \\ \Delta \text{ tight}}} \mathrm{R}(U_{\Delta,q})$$

and the same inequalities hold for the size of the tensors. For every tight T, we may find tight Δ such that $\sigma(T) \leq \sigma(\Delta)$. By Theorem 4 this equals $\sigma(\mathcal{U}_\Delta)$ and by the inequality above this is upper bounded by $\sigma(\mathcal{T}_d)$.

For the other inequality, we see that $\sigma(\mathcal{T}_d)$ is upper bounded by $\sigma(\mathcal{U}_\Delta)$ for some tight Δ. Clearly, $\sigma(\mathcal{U}_\Delta) \leq \sigma(\mathcal{T}_d)$ which finishes the proof. ◀

4 Representation Theory and Universality of Specht Tensors

Let us now take a representation-theoretic view to the invariant subspace $(\mathbb{C}^{[d]^p \times [d]^p \times [d]^p})^{\mathfrak{S}_p}$ and start by observing that the $\mathfrak{S}_p$-module $\mathbb{C}^{[d]^p \times [d]^p \times [d]^p}$ is isomorphic to the $\mathfrak{S}_p$-module $\mathbb{C}^{d^p} \otimes \mathbb{C}^{d^p} \otimes \mathbb{C}^{d^p}$. A k-tuple $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ of integers $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k \geq 1$ is an *(integer) partition* of the integer $p = \lambda_1 + \lambda_2 + \ldots + \lambda_k$ into k *parts*. We write $\mathcal{I}(p)$ for the set of all integer partitions of p and $\mathcal{I}(p, d)$ for the set of all integer partitions with at most d parts. For $\lambda \in \mathcal{I}(p)$ we write $\lambda! = (\lambda_1!)(\lambda_2!)\cdots(\lambda_k!)$. For $\lambda \in \mathcal{I}(p, d)$ we write $D(\lambda) = (d-k)! \prod_{\ell=1}^{p} (|\{j \in [k] : \lambda_j = \ell\}|!)$. For $\lambda \in \mathcal{I}(p)$, let us write M^λ for the $\mathfrak{S}_p$-permutation module on Young tabloids of shape λ. Recall that the dimension of M^λ is $p!/\lambda!$. We have the $\mathfrak{S}_p$-module isomorphism

$$\mathbb{C}^{d^p} \cong \bigoplus_{\lambda \in \mathcal{I}(p,d)} \frac{d!}{D(\lambda)} M^\lambda$$

and thus

$$\mathbb{C}^{d^p} \otimes \mathbb{C}^{d^p} \otimes \mathbb{C}^{d^p} \cong \bigoplus_{\mu,\nu,\lambda \in \mathcal{I}(p,d)} \frac{d!}{D(\mu)} \frac{d!}{D(\nu)} \frac{d!}{D(\lambda)} M^\mu \otimes M^\nu \otimes M^\lambda . \tag{9}$$

For $\alpha \in \mathcal{I}(p)$, let us write S^α for the Specht module associated with α. For $\alpha, \mu \in \mathcal{I}(p)$, let us write $m_{\alpha,\mu}$ for the Kostka numbers so that the $\mathfrak{S}_p$-module isomorphism

$$M^\mu \cong \bigoplus_{\alpha \in \mathcal{I}(p)} m_{\alpha,\mu} S^\alpha \tag{10}$$

holds. We have the following elementary lemma that puts a strong restriction on the Specht modules that need to be considered for small values of d in (9) and (10). Recall that a partition $(\alpha_1, \alpha_2, \dots)$ is greater or equal to a partition $(\mu_1, \mu_2, \dots)$ in the *dominance order* if and only if for any i, we have $\sum_{j=0}^{i} \alpha_j \geq \sum_{j=0}^{i} \mu_j$. Here, if one of partitions, say α, has k parts, we let $\alpha_j = 0$ for $j > k$.

▶ **Lemma 35** (Restriction to at most d parts)**.** *For $\alpha, \mu \in \mathcal{I}(p)$ we have $m_{\alpha,\mu} \neq 0$ if and only if α is larger than μ in the dominance order. In particular, for all $\mu \in \mathcal{I}(p,d)$ we have $m_{\alpha,\mu} = 0$ unless $\alpha \in \mathcal{I}(p,d)$.*

Proof. See e.g. [64, p. 315]. ◀

From (9), (10), and Lemma 35 we thus have

$$\mathbb{C}^{d^p} \otimes \mathbb{C}^{d^p} \otimes \mathbb{C}^{d^p} \cong \bigoplus_{\mu,\nu,\lambda \in \mathcal{I}(p,d)} \frac{d!}{D(\mu)} \frac{d!}{D(\nu)} \frac{d!}{D(\lambda)} \bigoplus_{\alpha,\beta,\gamma \in \mathcal{I}(p,d)} m_{\alpha,\mu} m_{\beta,\nu} m_{\gamma,\lambda} S^\alpha \otimes S^\beta \otimes S^\gamma \,. \tag{11}$$

Writing $K(\alpha,\beta,\gamma)$ for the Kronecker coefficient of $\alpha, \beta, \gamma \in \mathcal{I}(p)$, we have that the invariant space $(S^\alpha \otimes S^\beta \otimes S^\gamma)^{\mathfrak{S}_p}$ has dimension $K(\alpha,\beta,\gamma)$. Since $(\mathbb{C}^{[d]^p \times [d]^p \times [d]^p})^{\mathfrak{S}_p}$ has dimension $\binom{d^3-1+p}{d^3-1}$, from (11) we immediately have

$$\binom{d^3-1+p}{d^3-1} = \sum_{\mu,\nu,\lambda \in \mathcal{I}(p,d)} \frac{d!}{D(\mu)} \frac{d!}{D(\nu)} \frac{d!}{D(\lambda)} \sum_{\alpha,\beta,\gamma \in \mathcal{I}(p,d)} m_{\alpha,\mu} m_{\beta,\nu} m_{\gamma,\lambda} K(\alpha,\beta,\gamma) \,. \tag{12}$$

Assuming that d is fixed and p grows, we observe that the left-hand side of (12) grows at most polynomially in p, and all summands on the right-hand side are nonnegative, implying that every summand on the right-hand grows at most polynomially in p. The above observations motivate the following definition.

▶ **Definition 36.** *For any three Specht modules $S^\alpha, S^\beta, S^\gamma$, where α, β, γ are partitions of p, we call any element of $(S^\alpha \otimes S^\beta \otimes S^\gamma)^{\mathfrak{S}_p}$ an* invariant Specht tensor *with* margin α, β, γ *and* degree p*.*

▶ Remark 37. For fixed α, β, γ the dimension of the space of invariant Specht tensors equals the Kronecker coefficient $K(\alpha,\beta,\gamma)$.

By Lemma 35 and Equation (11) only invariant Specht tensors with margins that have at most d parts appear in the decomposition of $(\mathbb{C}^{[d]^p \times [d]^p \times [d]^p})^{\mathfrak{S}_p}$ and each such tensor does appear. Further, for fixed d the dimension of all invariant Specht tensors with arbitrary margins $\alpha, \beta, \gamma \in \mathcal{I}(p,d)$ is polynomial in p. Thus, we can equivalently define $\sigma(d)$ as the infimum of all positive real numbers such that for all $\alpha, \beta, \gamma \in \mathcal{I}(p,d)$ the maximum tensor rank in $(S^\alpha \otimes S^\beta \otimes S^\gamma)^{\mathfrak{S}_p}$ is at most $d^{\sigma(d)p+o(p)}$. We obtain the following corollary.

▶ **Corollary 38.** *Fix an integer $d \geq 2$. Then $\sigma(d)$ is the infimum of all positive real numbers such that for all $\alpha, \beta, \gamma \in \mathcal{I}(p,d)$ the maximum tensor rank of an invariant Specht tensor is at most $d^{\sigma(d)p+o(p)}$.*

The above corollary motivates the following questions:

- What is the maximum tensor rank for an invariant Specht tensor of degree p with margins in $\mathcal{I}(p,d)$ as p grows?
- Which margins witness this maximum?

Specht modules and tensor products of Specht modules have considerable structure; e.g. [77]. As highlighted above, already the case $d = 3$ is interesting.

▶ Remark 39. Our results allow to translate between asymptotic rank and rank of invariant Specht tensors. As $\sigma(d) \leq \frac{2\omega}{3} < 2$, we see that in the range $\alpha, \beta, \gamma \in \mathcal{I}(p,d)$ one indeed obtains nontrivial upper bounds on ranks of invaiant Specht tensors; that is, such tensors in general do not have generic ranks in their ambient spaces $S^\alpha \otimes S^\beta \otimes S^\gamma$.

5 Equations of Secant Varieties and Asymptotic Rank

In this section we relate existence of polynomials vanishing on special varieties, like secant varieties, and asymptotic rank. In particular, we obtain upper-bound control on $\sigma(d)$ via the *absence* of low-degree equations of secants.

5.1 Equations and Bounds on $\sigma(d)$

We start with a generalization of Proposition 15.

▶ **Proposition 40** (Span of the image of a subset). *For any subset $S \subseteq (\mathbb{F}^d)^{\otimes 3}$, the following are equivalent:*

1. *no homogeneous polynomial of degree p vanishes on S,*
2. *$K_{d,p}(S)$ linearly spans $L_{d,p}$.*

Proof. The second condition is equivalent to the fact that no linear form in $L_{d,p}^*$ vanishes on $K_{d,p}(S)$. By Lemma 14 this is also equivalent to the first condition. ◀

▶ **Corollary 41** (Absence of low-degree equations implies low asymptotic rank). *Let X be the Segre variety of rank one tensors in $(\mathbb{F}^d)^{\otimes 3}$. Suppose that for fixed n no polynomial of degree p vanishes on the n^{th} secant variety $\sigma_n(X)$. Then every tensor in $L_{d,p}$ has rank at most*

$$\binom{d^3-1+p}{d^3-1} n^p .$$

Every tensor in $(\mathbb{F}^d)^{\otimes 3}$ has asymptotic rank at most

$$n \binom{d^3-1+p}{d^3-1}^{\frac{1}{p}} .$$

Proof. Note that $\dim L_{d,p} = \binom{d^3-1+p}{d^3-1}$ and by Proposition 40 we have a basis of $L_{d,p}$ made of tensors of rank at most n^p. The last statement follows. ◀

A more general, but less explicit version is given by Theorem 8, which we derive as the following corollary.

▶ **Corollary 42** (Absence of low-degree equations implies low asymptotic rank; implicit version). *Let $Y \subseteq (\mathbb{F}^d)^{\otimes 3}$ be a subset with the property that for all $T \in Y$ the asymptotic rank of T is at most n. Suppose that no homogeneous polynomial of degree p vanishes on Y. Then, every tensor in $(\mathbb{F}^d)^{\otimes 3}$ has asymptotic rank at most*

$$n \binom{d^3-1+p}{d^3-1}^{\frac{1}{p}} .$$

▶ Remark 43. The dimension component $\binom{d^3-1+p}{d^3-1}$ may be improved (by going to border rank) by the smallest s, such that $\sigma_s(K_{d,p}(\sigma_n(X))) = L_{d,p}$. The latter number is in most cases much smaller then $\dim L_{d,p}$. Each particular case may be exactly studied by the Teraccini Lemma.

Alternatively we could greedily, starting from rank one and going up, take tensors T_i such that $K_{d,p}(T_i)$ are linearly independent, building a basis of $L_{d,p}$.

For tensors belonging to a fixed subspace $W \subseteq (\mathbb{F}^d)^{\otimes 3}$ we also have the following variant.

▶ **Theorem 44.** *Let $Y \subseteq W \subseteq (\mathbb{F}^d)^{\otimes 3}$ be a subset with the property that for all $T \in Y$ the asymptotic rank of T is at most n. Suppose that no homogeneous polynomial on W of degree p vanishes on Y. Then every tensor in W has asymptotic rank at most*

$$n\binom{\dim W - 1 + p}{\dim W - 1}^{\frac{1}{p}}.$$

Proof. We recall that the restriction of the Veronese map of degree p to any linear subspace is still a Veronese map of degree p; that is, a map defined by polynomials spanning the space of degree p polynomials, simply in smaller number of variables.

We restrict $K_{d,p}$ to W. The linear span of $K_{d,p}(W)$ is spanned by $K_{d,p}(Y)$ and has dimension $\binom{\dim W-1+p}{\dim W-1}$. Hence for any tensor $T \in W$ the asymptotic rank of $K_{d,p}(T)$ is at most:

$$n^p\binom{\dim W - 1 + p}{\dim W - 1}.$$

As $K_{d,p}(T)$ is the p^{th} Kronecker power of T, the statement follows. ◀

The previous result could be particularly useful if we can provide many examples of low rank tensors inside a linear subspace.

5.2 Equations of Secant Varieties and Applications

We note that deciding if there exists a nonzero polynomial of degree p on $(\mathbb{F}^d)^{\otimes 3}$ vanishing on tensors of rank r is a question in linear algebra. The naïve approach is a as follows.

Consider $3r$ vectors $v_{i,j} \in \mathbb{F}^d$ where $1 \leq i \leq 3$ and $1 \leq j \leq r$ with coordinates $(v_{i,j})_k$ treated as variables. The tensor $T := \sum_{j=1}^r v_{1,j} \otimes v_{2,j} \otimes v_{3,j}$ is a general tensor of rank r. We have:

$$T_{a,b,c} = \sum_{j=1}^{r} (v_{1,j})_a (v_{2,j})_b (v_{3,j})_c \,.$$

If we assign the weight $(1,0,0) \in \mathbb{Z}^3$ to each $(v_{1,j})_a$, the weight $(0,1,0)$ to each $(v_{2,j})_b$, and the weight $(0,0,1)$ to each $(v_{3,j})_c$, we see that an evaluation of a degree p polynomial P on T is a degree (p,p,p) polynomial. Furthermore, P vanishes on all tensors of rank at most r if and only if $P(T) = 0$ as a polynomial. To check the last condition we may build a matrix $N_{d,r,p}$ with:

1. $\binom{p+d^3-1}{p}$ columns indexed by $\mathcal{C}_p^{[d]^3}$, equivalently monomials of degree p,
2. $\binom{p+dr-1}{p}^3$ rows indexed by monomials of degree (p,p,p) in variables $(v_{i,j})_k$,
3. the entry in a column indexed by a monomial $g \in \mathcal{C}_p^{[d]^3}$ and a row indexed by a monomial m is the coefficient of m in the polynomial $g(T)$.

The kernel of $N_{d,r,p}$ is naturally identified with the space of homogeneous degree p polyomials vanishing on rank r tensors. In particular, the kernel is trivial if and only if there are no such polynomials. While this approach is very general, it is also not applicable in most cases due to the large sizes of the matrices $N_{d,r,p}$.

A better approach to understand the equations of secant varieties is to exploit group actions. For this, one decomposes the space $S^p(\mathbb{C}^d \otimes \mathbb{C}^d \otimes \mathbb{C}^d)$ of homogeneous degree p polynomials on the tensor space as a direct sum of irreducible $G := GL(n) \times GL(n) \times GL(n)$ representations. The irreducible polynomial representations of G are indexed by triples of Young diagrams, in our case each one in the triple will have p elements:

$$S^p(\mathbb{C}^d \otimes \mathbb{C}^d \otimes \mathbb{C}^d) = \bigoplus V_{\lambda,\mu,\nu}^{\bigoplus g_{\lambda,\mu,\nu}},$$

where $V_{\lambda,\mu,\nu}$ is the triple corresponding to three Young diagrams λ, μ, ν with p boxes each and $V_{\lambda,\mu,\nu}$ is the Kronecker coefficient. The vector space of homogeneous polynomials of degree p that vanish on the n^{th} secant variety is a subrepresentation and hence also may be decomposed as:

$$\bigoplus V_{\lambda,\mu,\nu}^{\bigoplus a_{\lambda,\mu,\nu}},$$

where now the coefficients $a_{\lambda,\mu,\nu} \leq g_{\lambda,\mu,\nu}$ are unknown. One way to determine them is to consider highest weigh space $HWS_{\lambda,\mu,\nu}$, that is a vector space of dimension $g_{\lambda,\mu,\nu}$ inside $V_{\lambda,\mu,\nu}^{\bigoplus g_{\lambda,\mu,\nu}}$. It turns out that $a_{\lambda,\mu,\nu}$ equals the dimension of the subspace of $HWS_{\lambda,\mu,\nu}$ consisting on those polynomials that vanish on the n^{th} secant variety. Finding the basis of $HWS_{\lambda,\mu,\nu}$ and efficiently evaluating it on points of the secant variety is an art on its own. However, in many examples, the procedure above was successfully carried out in practice. We refer to [11, 36] for details.

▶ **Example 45.** The generic rank in $\mathbb{C}^7 \otimes \mathbb{C}^7 \otimes \mathbb{C}^7$ is 19. Tensors of border rank 18 form a hypersurface with the defining equation of degree at least 187000 [36, Section 3.2]. Applying Corollary 41 we see that every tensor in that space has asymptotic rank smaller than 18.25.

We believe that many further results on asymptotic rank may be obtained in a similar way, through Corollary 41 and 42, especially in combination with Remark 43. We leave this for future work.

References

1 Boris Alexeev, Michael A. Forbes, and Jacob Tsimerman. Tensor rank: Some lower and upper bounds. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, USA, June 8-10, 2011*, pages 283–291. IEEE Computer Society, 2011. doi:10.1109/CCC.2011.28.

2 Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 522–539. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2021. doi:10.1137/1.9781611976465.32.

3 Josh Alman and Hengjie Zhang. Generalizations of matrix multiplication can solve the light bulb problem. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 1471–1495. IEEE, 2023. doi:10.1109/FOCS57990.2023.00090.

4 Alessandra Bernardi and Kristian Ranestad. On the cactus rank of cubics forms. *J. Symbolic Comput.*, 50:291–297, 2013. doi:10.1016/j.jsc.2012.08.001.

5 Alessandra Bernardi and Daniele Taufer. Waring, tangential and cactus decompositions. *J. Math. Pures Appl. (9)*, 143:1–30, 2020. doi:10.1016/j.matpur.2020.07.003.

6 Dario Bini, Milvio Capovani, Francesco Romani, and Grazia Lotti. $O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication. *Inform. Process. Lett.*, 8(5):234–235, 1979. doi:10.1016/0020-0190(79)90113-3.

7 Andreas Björklund and Petteri Kaski. The asymptotic rank conjecture and the set cover conjecture are not both true. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 859–870. ACM, 2024. doi:10.1145/3618260.3649656.

8 Markus Bläser. A $\frac{5}{2}n^2$-lower bound for the rank of $n \times n$-matrix multiplication over arbitrary fields. In *40th Annual Symposium on Foundations of Computer Science (New York, 1999)*, pages 45–50. IEEE Computer Soc., Los Alamitos, CA, 1999. doi:10.1109/SFFCS.1999.814576.

9 Markus Bläser, Matthias Christandl, and Jeroen Zuiddam. The border support rank of two-by-two matrix multiplication is seven. *Chic. J. Theoret. Comput. Sci.*, pages Art. 5, 16, 2018. doi:10.4086/cjtcs.2018.005.

10 Jonah Blasiak, Henry Cohn, Joshua A. Grochow, Kevin Pratt, and Chris Umans. Matrix multiplication via matrix groups. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 19:1–19:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.ITCS.2023.19.

11 Paul Breiding, Reuven Hodges, Christian Ikenmeyer, and Mateusz Michałek. Equations for GL invariant families of polynomials. *Vietnam J. Math.*, 50(2):545–556, 2022. doi:10.1007/s10013-022-00549-4.

12 Weronika Buczyńska and Jarosław Buczyński. Secant varieties to high degree Veronese reembeddings, catalecticant matrices and smoothable Gorenstein schemes. *J. Algebraic Geom.*, 23(1):63–90, 2014. doi:10.1090/S1056-3911-2013-00595-0.

13 Weronika Buczyńska and Jarosław Buczyński. Apolarity, border rank, and multigraded Hilbert scheme. *Duke Math. J.*, 170(16):3659–3702, 2021. doi:10.1215/00127094-2021-0048.

14 Jarosław Buczyński and J. M. Landsberg. Ranks of tensors and a generalization of secant varieties. *Linear Algebra Appl.*, 438(2):668–689, 2013. doi:10.1016/j.laa.2012.05.001.

15 Peter Bürgisser. *Degenerationsordnung und Trägerfunktional bilinearer Abbildungen*. PhD thesis, Universität Konstanz, 1990. URL: http://nbn-resolving.de/urn:nbn:de:bsz:352-opus-20311.

16 Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1997. doi:10.1007/978-3-662-03338-8.

17 Matthias Christandl, Péter Vrana, and Jeroen Zuiddam. Barriers for fast matrix multiplication from irreversibility. *Theory Comput.*, 17:Paper No. 2, 32, 2021. doi:10.4086/toc.2021.v017a002.

18 Matthias Christandl, Péter Vrana, and Jeroen Zuiddam. Universal points in the asymptotic spectrum of tensors. *J. Amer. Math. Soc.*, 36(1):31–79, 2023. doi:10.1090/jams/996.

19 Henry Cohn, Robert D. Kleinberg, Balázs Szegedy, and Christopher Umans. Group-theoretic algorithms for matrix multiplication. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 379–388. IEEE Computer Society, 2005. doi:10.1109/SFCS.2005.39.

20 Henry Cohn and Christopher Umans. A group-theoretic approach to fast matrix multiplication. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 438–449. IEEE Computer Society, 2003. doi:10.1109/SFCS.2003.1238217.

21 Henry Cohn and Christopher Umans. Fast matrix multiplication using coherent configurations. In Sanjeev Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1074–1087. SIAM, 2013. doi:10.1137/1.9781611973105.77.

22 Austin Conner, Fulvio Gesmundo, Joseph M. Landsberg, and Emanuele Ventura. Rank and border rank of Kronecker powers of tensors and Strassen's laser method. *Comput. Complexity*, 31(1):Paper No. 1, 40, 2022. doi:10.1007/s00037-021-00217-y.

23 Austin Conner, Fulvio Gesmundo, Joseph M. Landsberg, Emanuele Ventura, and Yao Wang. Towards a geometric approach to Strassen's asymptotic rank conjecture. *Collect. Math.*, 72(1):63–86, 2021. doi:10.1007/s13348-020-00280-8.

24 Austin Conner, Alicia Harper, and J. M. Landsberg. New lower bounds for matrix multiplication and $\det_3$. *Forum Math. Pi*, 11:Paper No. e17, 30, 2023. doi:10.1017/fmp.2023.14.

25 D. Coppersmith and S. Winograd. On the asymptotic complexity of matrix multiplication. *SIAM J. Comput.*, 11(3):472–492, 1982. doi:10.1137/0211038.

26 Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *J. Symbolic Comput.*, 9(3):251–280, 1990. doi:10.1016/S0747-7171(08)80013-2.

27 David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms*. Undergraduate Texts in Mathematics. Springer, Cham, fourth edition, 2015. doi:10.1007/978-3-319-16721-3.

28 Imre Csiszár and János Körner. *Information Theory*. Cambridge University Press, Cambridge, second edition, 2016.

29 Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström. On problems as hard as CNF-SAT. *ACM Trans. Algorithms*, 12(3):41:1–41:24, 2016. doi:10.1145/2925416.

30 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

31 Ran Duan, Hongxun Wu, and Renfei Zhou. Faster matrix multiplication via asymmetric hashing. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 2129–2138. IEEE, 2023. doi:10.1109/FOCS57990.2023.00130.

32 Maciej Gałązka. Vector bundles give equations of cactus varieties. *Linear Algebra Appl.*, 521:254–262, 2017. doi:10.1016/j.laa.2016.12.005.

33 Maciej Gałązka. Multigraded apolarity. *Math. Nachr.*, 296(1):286–313, 2023. doi:10.1002/mana.202000484.

34 Philip Alan Gartenberg. *Fast Rectangular Matrix Multiplication*. PhD thesis, University of California, Los Angeles, 1985. URL: https://www.proquest.com/dissertations-theses/fast-rectangular-matrix-multiplication-algebraic/docview/303332114/se-2.

35 Johan Håstad. Tensor rank is NP-complete. *J. Algorithms*, 11(4):644–654, 1990. doi:10.1016/0196-6774(90)90014-6.

36 Jonathan D. Hauenstein, Christian Ikenmeyer, and J. M. Landsberg. Equations for lower bounds on border rank. *Exp. Math.*, 22(4):372–383, 2013. doi:10.1080/10586458.2013.825892.

37 Christopher J. Hillar and Lek-Heng Lim. Most tensor problems are NP-hard. *J. ACM*, 60(6):Art. 45, 39, 2013. doi:10.1145/2512329.

38 Anthony Iarrobino and Vassil Kanev. *Power Sums, Gorenstein Algebras, and Determinantal Loci*, volume 1721 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1999. doi:10.1007/BFb0093426.

39 Matti Karppa and Petteri Kaski. Probabilistic tensors and opportunistic Boolean matrix multiplication. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 496–515. SIAM, Philadelphia, PA, 2019. doi:10.1137/1.9781611975482.31.

40 Robert Krauthgamer and Ohad Trabelsi. The set cover conjecture and subgraph isomorphism with a tree pattern. In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, volume 126 of *LIPIcs*, pages 45:1–45:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.STACS.2019.45.

41 J. M. Landsberg. The border rank of the multiplication of 2×2 matrices is seven. *J. Amer. Math. Soc.*, 19(2):447–459, 2006. doi:10.1090/S0894-0347-05-00506-0.

42 J. M. Landsberg. *Tensors: Geometry and Applications*, volume 128 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2012. doi:10.1090/gsm/128.

43 J. M. Landsberg. New lower bounds for the rank of matrix multiplication. *SIAM J. Comput.*, 43(1):144–149, 2014. doi:10.1137/120880276.

44 J. M. Landsberg. Nontriviality of equations and explicit tensors in $\mathbb{C}^m \otimes \mathbb{C}^m \otimes \mathbb{C}^m$ of border rank at least $2m - 2$. *J. Pure Appl. Algebra*, 219(8):3677–3684, 2015. doi:10.1016/j.jpaa.2014.12.016.

45 J. M. Landsberg. *Tensors: Asymptotic Geometry and Developments 2016–2018*, volume 132 of *CBMS Regional Conference Series in Mathematics*. American Mathematical Society, Providence, RI, 2019. doi:10.1090/cbms/132.

46 J. M. Landsberg and L. Manivel. On the ideals of secant varieties of Segre varieties. *Found. Comput. Math.*, 4(4):397–422, 2004. doi:10.1007/s10208-003-0115-9.

47 J. M. Landsberg and Mateusz Michałek. On the geometry of border rank decompositions for matrix multiplication and other tensors with symmetry. *SIAM J. Appl. Algebra Geom.*, 1(1):2–19, 2017. doi:10.1137/16M1067457.

48 J. M. Landsberg and Mateusz Michałek. Towards finding hay in a haystack: explicit tensors of border rank greater than $2.02m$ in $\mathbb{C}^m \otimes \mathbb{C}^m \otimes \mathbb{C}^m$. *CoRR*, abs/1912.11927, 2019. doi:10.48550/arXiv.1912.11927.

49 J. M. Landsberg and Giorgio Ottaviani. Equations for secant varieties of Veronese and other varieties. *Ann. Mat. Pura Appl. (4)*, 192(4):569–606, 2013. doi:10.1007/s10231-011-0238-6.

50 J. M. Landsberg and Zach Teitler. On the ranks and border ranks of symmetric tensors. *Found. Comput. Math.*, 10(3):339–366, 2010. doi:10.1007/s10208-009-9055-3.

51 Joseph M. Landsberg and Mateusz Michałek. A $2n^2 - \log_2(n) - 1$ lower bound for the border rank of matrix multiplication. *Int. Math. Res. Not. IMRN*, 15:4722–4733, 2018. doi:10.1093/imrn/rnx025.

52 Joseph M. Landsberg and Giorgio Ottaviani. New lower bounds for the border rank of matrix multiplication. *Theory Comput.*, 11:285–298, 2015. doi:10.4086/toc.2015.v011a011.

53 François Le Gall. Powers of tensors and fast matrix multiplication. In *ISSAC 2014—Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, pages 296–303. ACM, New York, 2014. doi:10.1145/2608628.2608664.

54 Franz Mauch. *Ein Randverteilungsproblem und seine Anwendung auf das asymptotische Spektrum bilinearer Abbildungen*. PhD thesis, Universität Konstanz, 1998. URL: http://nbn-resolving.de/urn:nbn:de:bsz:352-opus-20543.

55 Mateusz Michałek and Bernd Sturmfels. *Invitation to Nonlinear Algebra*, volume 211 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2021. doi:10.1090/gsm/211.

56 Ketan Mulmuley. The GCT program toward the P vs. NP problem. *Commun. ACM*, 55(6):98–107, 2012. doi:10.1145/2184319.2184341.

57 Ketan D. Mulmuley. On P vs. NP and geometric complexity theory. *J. ACM*, 58(2):Art. 5, 26, 2011. doi:10.1145/1944345.1944346.

58 Victor Y. Pan. Strassen's algorithm is not optimal: Trililnear technique of aggregating, uniting and canceling for constructing fast algorithms for matrix operations. In *19th Annual Symposium on Foundations of Computer Science, Ann Arbor, Michigan, USA, 16-18 October 1978*, pages 166–176. IEEE Computer Society, 1978. doi:10.1109/SFCS.1978.34.

59 Kevin Pratt. A stronger connection between the asymptotic rank conjecture and the set cover conjecture. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 871–874. ACM, 2024. doi:10.1145/3618260.3649620.

60 Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. *J. ACM*, 60(6):Art. 40, 15, 2013. doi:10.1145/2535928.

61 Francesco Romani. Some properties of disjoint sums of tensors related to matrix multiplication. *SIAM J. Comput.*, 11(2):263–267, 1982. doi:10.1137/0211020.

62 A. Schönhage. Partial and total matrix multiplication. *SIAM J. Comput.*, 10(3):434–455, 1981. doi:10.1137/0210032.

63 Zhao Song, David P. Woodruff, and Peilin Zhong. Relative error tensor low rank approximation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2772–2789. SIAM, Philadelphia, PA, 2019. doi:10.1137/1.9781611975482.172.

64 Richard P. Stanley. *Enumerative Combinatorics. Vol. 2*, volume 62 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1999. doi:10.1017/CBO9780511609589.

65 Andrew James Stothers. *On the Complexity of Matrix Multiplication*. PhD thesis, University of Edinburgh, 2010. URL: http://hdl.handle.net/1842/4734.

66 V. Strassen. Relative bilinear complexity and matrix multiplication. *J. Reine Angew. Math.*, 375/376:406–443, 1987. doi:10.1515/crll.1987.375-376.406.

67 V. Strassen. The asymptotic spectrum of tensors. *J. Reine Angew. Math.*, 384:102–152, 1988. doi:10.1515/crll.1988.384.102.

68 V. Strassen. Degeneration and complexity of bilinear maps: some asymptotic spectra. *J. Reine Angew. Math.*, 413:127–180, 1991. doi:10.1515/crll.1991.413.127.

69 V. Strassen. Algebra and complexity. In *First European Congress of Mathematics, Vol. II (Paris, 1992)*, volume 120 of *Progr. Math.*, pages 429–446. Birkhäuser, Basel, 1994.

70 Volker Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13:354–356, 1969. doi:10.1007/BF02165411.

71 Volker Strassen. The asymptotic spectrum of tensors and the exponent of matrix multiplication. In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 49–54. IEEE Computer Society, 1986. doi:10.1109/SFCS.1986.52.

72 Volker Strassen. Komplexität und Geometrie bilinearer Abbildungen. *Jahresber. Deutsch. Math.-Verein.*, 107(1):3–31, 2005.

73 Zach Teitler. Geometric lower bounds for generalized ranks. *CoRR*, abs/1406.5145, 2014. doi:10.48550/arXiv.1406.5145.

74 Verena Tobler. *Spezialisierung und Degeneration von Tensoren*. PhD thesis, Universität Konstanz, 1991. URL: http://nbn-resolving.de/urn:nbn:de:bsz:352-opus-20324.

75 Virginia Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd [extended abstract]. In *STOC'12—Proceedings of the 2012 ACM Symposium on Theory of Computing*, pages 887–898. ACM, New York, 2012. doi:10.1145/2213977.2214056.

76 Avi Wigderson and Jeroen Zuiddam. Asymptotic spectra: Theory, applications and extensions. Manuscript dated October 24, 2023; available at https://www.math.ias.edu/~avi/PUBLICATIONS/WigdersonZu_Final_Draft_Oct2023.pdf, 2023.

77 John D. Wiltshire-Gordon, Alexander Woo, and Magdalena Zajaczkowska. Specht polytopes and Specht matroids. In *Combinatorial Algebraic Geometry*, volume 80 of *Fields Inst. Commun.*, pages 201–228. Fields Inst. Res. Math. Sci., Toronto, ON, 2017.

78 F. L. Zak. *Tangents and Secants of Algebraic Varieties*, volume 127 of *Translations of Mathematical Monographs*. American Mathematical Society, Providence, RI, 1993. doi:10.1090/mmono/127.

Online Versus Offline Adversaries in Property Testing

Esty Kelman
Boston University, MA, USA
Massachusetts Institute of Technology, Cambridge, MA, USA

Ephraim Linder
Boston University, MA, USA

Sofya Raskhodnikova
Boston University, MA, USA

Abstract

We study property testing with incomplete or noisy inputs. The models we consider allow for *adversarial* manipulation of the input, but differ in whether the manipulation can be done only *offline*, i.e., before the execution of the algorithm, or *online*, i.e., as the algorithm runs. The manipulations by an adversary can come in the form of erasures or corruptions. We compare the query complexity and the randomness complexity of property testing in the offline and online models. Kalemaj, Raskhodnikova, and Varma (Theory Comput. '23) provide properties that can be tested with a small number of queries with offline erasures, but cannot be tested at all with online erasures. We demonstrate that the two models are incomparable in terms of query complexity: we construct properties that can be tested with a constant number of queries in the online corruption model, but require querying a significant fraction of the input in the offline erasure model. We also construct properties that exhibit a strong separation between the randomness complexity of testing in the presence of offline and online adversaries: testing these properties in the online model requires exponentially more random bits than in the offline model, even when they are tested with nearly the same number of queries in both models. Our randomness separation relies on a novel reduction from randomness-efficient testers in the adversarial online model to query-efficient testers in the standard model.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Property Testing, Online Adversary, Offline Adversary, Query Complexity, Randomness Complexity, Separations

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.65

Related Version *Full Version*: `https://arxiv.org/abs/2411.18617` [22]

Funding *Esty Kelman*: Supported in part by the National Science Foundation under Grant No. 2022446 and in part by NSF TRIPODS program (award DMS-2022448).

Acknowledgements The authors thank Uri Meir for fruitful discussions regarding the online model.

1 Introduction

Property testing [31, 15] aims to lay algorithmic foundations for processing big data. It is a formal study of fast algorithms that accept objects with a given property and reject objects that are far. The goal of this work is to compare models of property testing that address the situations when the data that needs to be analyzed is not only large, but also incomplete or noisy. The models we consider allow for adversarial manipulation of the input, but differ in whether the manipulation can be done only *offline*, i.e., before the execution of the algorithm, or *online*, i.e., as the algorithm runs. The manipulations by an adversary can come in the


© Esty Kelman, Ephraim Linder, and Sofya Raskhodnikova;
licensed under Creative Commons License CC-BY 4.0



form of erasures or corruptions. The offline erasure-resilient property testing model was proposed by Dixit et al. [9]. The model that captures offline corruptions is called *tolerant testing* and was introduced by Parnas et al. [25]. The online analogues of these models were recently defined by Kalemaj et al. [21].

We compare the query complexity and the randomness complexity of property testing in the offline and online models. In the offline-erasure model, a constant fraction of the input is erased adversarially before the execution of the algorithm, whereas in the online-erasure model, the adversary is allowed to make a fixed number of erasures after each query. It may seem that the online adversary has strictly more power, and thus testing in the online models should require more queries. Indeed, [21] justify this intuition and show that simple properties – sortedness and the Lipschitz property of sequences – cannot be tested at all with online erasures[1], even though they are testable using a small number of queries when erasures are offline. Intuitively, [21] show that testing in the presence of one online erasure per query (performed by an adversary with the knowledge of previous queries) can be harder than testing when a constant fraction of the input is adversarially erased offline (in advance). The following natural open question is stated in [21]: "Is there a property that has smaller query complexity in the online-erasure-resilient model than in the (offline) erasure-resilient model of [9]?" We answer this question in the affirmative: specifically, we construct a property that can be tested with a constant number of queries in the online-erasure model (and even in the (harder) online-corruption model), but requires a nearly linear number of queries in the offline-erasure model. We conclude that the two adversarial erasure models are incomparable.

Our second result concerns the randomness complexity of testing in the online and offline manipulation models. Randomness is an important tool in the design of a wide variety of algorithms, and extensive research has been conducted to understand the power of randomness in computation. The investigation of the randomness complexity of property testing algorithms was initiated by Goldreich and Sheffet [18]. They showed that all property testers in the standard model can be derandomized to a large extent: they can be converted to testers that use the number of random bits that is logarithmic in the size of the input, while incurring at most a constant factor blowup in query complexity. The result of [18] easily extends to all offline testing models, but does not apply in the online setting. Moreover, the existing testers in the online model [21, 24, 6, 33, 3] use lots of randomness to fool the adversary. A natural question that arises is how much derandomization is possible in the online setting. We show that a lot of randomness is sometimes indeed necessary to test in the presence of an online adversary. In particular, we exhibit a property that requires exponentially more random bits to test in the online manipulation model than in the offline manipulation model. This result provides a natural converse to the statement of [24] that "the (online) adversary cannot foil our plan if there is no plan" – i.e., random queries are robust to online manipulation of the input.

1.1 Our results

We represent the input to our property testing algorithm as a string. The algorithm is given access to its input string via queries (specifically, of the form "what is the character at index i?"). For each proximity parameter $\varepsilon \in (0, 1)$, the algorithm has to distinguish between strings that have a specified property and strings that are ε-far from having the property – i.e., strings that need to be changed on at least an ε-fraction of indices in order to satisfy

[1] In contrast, in the offline models, testing is always possible when the entire input is read.

the property. The query complexity (the number of queries to the input) of the tester is measured in terms of ε and the input length n. In this section, for simplicity, ε is considered to be a constant and omitted from the statements.

1.1.1 Query complexity: Model Incomparability

We show that the online and offline adversarial models of property testing are incomparable in terms of the query complexity. Recall that [21] prove that sortedness and Lipschitzness of sequences are efficiently testable with offline erasures, but cannot be tested at all even with one online erasure per query[2]. We complement this result with the following theorems, which show that testing in the presence of a small fraction of offline erasures can be harder than testing with online erasures. The first of this theorems (Theorem 1.1) considers the regime with many online erasures per query; the second (Theorem 1.1) considers the regime with one erasure per query. The former exhibits a logarithmic gap in query complexity, where as the latter exhibits a nearly linear gap.

▶ **Theorem 1.1** (Informal version of Corollary 3.2)**.** *There exists a property that can be tested using a* constant number of queries *in the presence of an online adversary that makes $O\left(\frac{n}{\log n}\right)$ erasures per query; but requires $\widetilde{\Omega}(\log n)$* queries[3] *to test when the characters at $\Theta\left(\frac{n}{\log\log n}\right)$ indices are erased offline (in advance).*

We cannot expect to prove an analogue of Theorem 1.1 for the case when the number of online erasures per query is the same as the total number of offline erasures. This is because, after the first query, an online adversary can erase the same part of the input that the offline adversary would have erased in advance.

▶ Remark 1.2. Although the online erasure rate in Theorem 1.1 is very large, the total number of erasures available to the online adversary is smaller than the number of offline erasures we considered. It is an intriguing open question whether there exists a property for which every tester in the online model accumulates at least as many erasures as the best tester in the offline model, but also requires fewer queries to test in the online model than in the offline model.

At the other end of the spectrum, we show that if the number of erasures is much smaller in the online model than in the offline model, then there is an exponentially large gap – i.e., there are properties that can be tested with a constant number of queries when the erasures are online but require a nearly linear number of queries in the offline erasure model.

▶ **Theorem 1.3** (Informal version of Corollary 3.3)**.** *There exists a property that can be tested with a constant number of queries in the presence of an online adversary that makes one erasure per query, but requires $\widetilde{\Omega}(n)$ queries to test when the characters at $\Theta\left(\frac{n}{\log n}\right)$ indices are erased offline (in advance).*

---

[2] We note that there is an even simpler property than sortedness and Lipschitzness of sequences that is easily testable with offline erasures or corruptions, but is not testable at all with online erasures. Moreover, this property is over the alphabet $\Sigma = \{0,1\}$. Specifically, $\mathcal{P} = \{ww : w \in \{0,1\}^*\}$. We can estimate the distance to this property within additive error ε using $O(\frac{1}{\varepsilon^2})$ queries (and therefore it is easily tolerantly testable). However, we cannot test it with online erasures because, whenever the tester queries some position p in an input string of length $2n$, the adversary can erase the corresponding position ($n+p$ if $p \le n$ and $p-n$ if $p > n$).

[3] The notation $\widetilde{\Omega}(f(n))$ hides polylog $f(n)$ factors, e.g., $\widetilde{\Omega}(\log n) = \Omega(\log n/\operatorname{polyloglog} n)$.

In fact, the query complexity separations in Theorems 1.1 and 1.3 hold even if the online adversary can make *corruptions* instead of erasures.

To prove Theorems 1.1 and 1.3, we show that any property that separates the offline model from the standard one, i.e., a property that is easy to test in the standard model but hard in the offline model, can be "lifted" to separate the offline and online models. We can then leverage the state of the art separation (between the standard and offline-manipulation testing models) of [5] to obtain strong separations between the online and offline manipulation models.

Our "lifting" operation is simply encoding the property with a repetition code. Intuitively, repetition won't make the property easier to test in the presence of offline erasures since the same erasures can be made on all copies. Our main technical proof for this part is to show that repetition code is robust against online adversaries. We prove the following general result.

▶ **Lemma 1.4** (Informal version of Lemma 3.5)**.** *If a property $\mathcal{P}$ is testable when no adversary is present then the repeated version $\mathcal{P}^r$, given by concatenating each string in $\mathcal{P}$ with itself r times, is testable in the presence of an online adversary (as long as the number of erasures per query is not too large).*

1.1.2 Randomness complexity separations

As discussed earlier, the investigation of the randomness complexity of property testing algorithms was initiated by Goldreich and Sheffet [18] who showed that every property that is testable in the standard model using q queries can be tested using $O(q)$ queries and $O(\log n)$ random bits. We build on their investigation by studying the randomness complexity of the erasure models. Though their result easily extends to the offline models, the same extension does not work in the online model. Indeed, we show that the derandomization of [18] cannot be extended to the online testing model – that is, we construct a property that requires $\omega(\log n)$ random bits to test in the online model.

Note that a randomness separation trivially holds for properties which are not testable with online erasures, or when the offline-erasure-resilient tester is allowed to query the entire input[4]. Prior to our work it was not clear if there is a randomness separation that applies to properties that are testable in the online model, and still holds if we require that the testers in both models to have a sublinear (in n) query complexity.

We demonstrate such a separation. In particular, we show that there exists a property that is testable in both models using a sublinear number of queries, and, moreover, testing this property in the online erasure model requires exponentially more random bits than testing it in the offline erasure model.

▶ **Theorem 1.5** (Informal version of Corollary 4.2)**.** *There exists a property that can be tested using $O(\sqrt{n})$ queries in the presence of either an online or an offline adversary. The tester against the offline adversary uses $O(\log n)$ random bits, whereas every tester in the presence of an online adversary that makes one erasure per query requires $\widetilde{\Omega}(\sqrt{n})$ random bits to succeed with constant probability.*

Theorem 1.5 follows easily from a more general result, Theorem 4.1, which we prove in Section 4 and which yields meaningful separations for larger online erasure rates as well. The main technique we develop to prove a lower bound on the randomness complexity of testing

[4] In this case, the tester in the offline model is deterministic. Note that it is impossible to deterministically test any nontrivial property in the online model, since the adversary can erase all but the first query made by a deterministic tester.

with an online adversary is a transformation from randomness-efficient testers in the online model to query efficient testers in the standard model. A special case of the guarantees of our transformation is stated in Lemma 1.6. (The general version appears in Lemma 4.3.)

To prove Theorem 4.1, we combine our transformation with existing results regarding the property of strings called τ-Distinct-Elements, which is parametrized by $\tau \in \mathbb{N}$, and consists of strings that have at most τ distinct characters. Testing τ-Distinct-Elements was recently investigated in [17, 2, 1, 20], and it is a natural testing version of the distinct elements approximation problem studied in [28, 32]. For a more detailed exposition regarding the history of the τ-Distinct-Elements property and its variants see [20].

▶ **Lemma 1.6** (Special case of Lemma 4.3 for one online erasure per query)**.** *Fix $r \in \mathbb{N}$. If a property $\mathcal{P}$ can be tested using r random bits in the presence of an online adversary that makes one erasure per query, then $\mathcal{P}$ can be tested in the standard model (with the same proximity parameter ε) using at most r queries.*

The technique from Lemma 1.6, combined with the derandomization results of [18], immediately yields a separation for $\varepsilon = o(\frac{1}{\log n})$: Consider the property $\mathcal{P} \subseteq \{0,1\}^n$ of being the zero string. By the folklore property testing bound, $\Theta(\frac{1}{\varepsilon})$ queries are necessary and sufficient to test $\mathcal{P}$. By [18], there exists a tester in the offline erasure model that uses $O(\log n)$ random bits. However, by Lemma 1.6, every tester in the online model must use $\Omega(\frac{1}{\varepsilon}) = \omega(\log n)$ random bits. While this works for small ε, the separation we show in Theorem 1.5 (and the more general Theorem 4.1) holds for constant ε as well.

1.2 Related work

Relationship between offline and standard models

In terms of query complexity, testing with offline erasures lies between standard and tolerant testing. By definition, an offline-erasure-resilient tester is also a property tester in the standard model, because it has to work on the inputs with no erasures. Not surprisingly, offline testing with erasures is no harder than offline testing with corruptions. This is formalized in [9, Theorem 1.4] that shows that the existence of a tolerant tester for a property implies the existence of an offline-erasure-resilient tester for that property for a comparable setting of parameters. There are also separations that show that standard testing is strictly easier than offline-erasure-resilient testing, which in turn is strictly easier than tolerant testing. Specifically, Dixit et al. [9] showed that a property defined by Fischer and Fortnow [13] can be tested in the standard model with a constant number of queries, but requires $n^{\Omega(1)}$ queries in the offline-erasure-resilient model. This separation was strengthened by Ben-Eliezer et al. [5], improving $n^{\Omega(1)}$ to $\widetilde{\Omega}(n)$. Finally, Raskhodnikova et al. [29] prove a separation similar to that of [9] between offline-erasure-resilient testing and tolerant testing. Thus, at a high level, we have a complete picture in terms of the relative difficulty of testing in the three offline models.

Relationship between offline and online testing

Kalemaj et al. [21] exhibit two properties of strings for which testing with online erasures is impossible for every proximity parameter $\varepsilon \leq \frac{1}{2}$, whereas testing in the offline models is easy. The first property, *sortedness*, consists of sorted arrays, i.e., strings x of length n such that $x_i \leq x_{i+1}$ for all $i \in [n-1]$. The query complexity of testing sortedness (for constant ε) is $\Theta(\log n)$ in both the standard and the offline erasures model (when the erased part is at most a constant fraction of the input) [11, 10, 26, 12, 7, 8, 27, 4, 9]. The second property, Lipschitzness, consists of strings $x \in \{0,1,2\}^n$ satisfying $|x_i - x_{i+1}| \leq 1$ for all $i \in [n-1]$. It can be tested with $\Theta(\frac{1}{\varepsilon})$ queries in both the standard and the offline models [19, 9].

2 Preliminaries

Notation

We use $[n]$ to denote the set of integers $\{1, 2, \ldots, n\}$, and $\mathbb{N}$ to denote the set of positive integers.

Property testing

We start by stating standard property testing definitions. We represent an input to a property testing algorithm as a string of length n over some alphabet Σ_n that might depend on n. For example, Σ_n might be $\{0, 1\}$ or $[n]$.

▶ **Definition 2.1** (Relative Hamming distance, property, ε-far)**.** *The* relative Hamming distance *between two strings x, y of length n is $\delta_H(x, y) = \Pr_{i\sim[n]}[x_i \neq y_i]$ where i is uniformly random index from $[n]$. For a set $\mathcal{S}$ of strings of length n, define $\delta_H(x, \mathcal{S}) = \min_{y\in\mathcal{S}} \delta_H(x, y)$. A property $\mathcal{P}$ is a subset of Σ^* given by $\bigcup_{n\in\mathbb{N}} \mathcal{P}_n$, where each $\mathcal{P}_n$ consists of strings of length n over some alphabet Σ_n. A string x of length n is* ε-far *from $\mathcal{P}$ if $\delta_H(x, \mathcal{P}_n) \geq \varepsilon$.*

▶ **Definition 2.2** (ε-tester in the standard model [31, 15])**.** *For every property $\mathcal{P} \subseteq \Sigma^*$ and proximity parameter $\varepsilon \in (0, 1)$, an algorithm $\mathcal{T}$ is an* ε-tester *for $\mathcal{P}$ if, given a parameter $n \in \mathbb{N}$ and oracle access to input $x \in \Sigma^n$, the algorithm $\mathcal{T}$ accepts with probability at least $\frac{2}{3}$ whenever $x \in \mathcal{P}$ and rejects with probability at least $\frac{2}{3}$ whenever x is ε-far from $\mathcal{P}$. A tester has a one-sided error if it always accepts inputs $x \in \mathcal{P}$.*

Offline manipulations

The offline model with erasures was introduced by [9] and subsequently studied in [30, 29, 5, 23]. The definition we use here is adapted from [23] and only differs from the original definition in how the parameter ε is interpreted. We use $\perp$ to denote an erased symbol in the input.

▶ **Definition 2.3** (α-erased string, completion)**.** *For each $\alpha \in (0, 1)$, a string $x \in (\Sigma \cup \{\perp\})^n$ is α-erased if at most an α fraction of symbols in it are $\perp$. The indices of the $\perp$ symbols in the string are called* erased*. A string $y \in \Sigma^n$ that differs from a string $x \in (\Sigma \cup \{\perp\})^n$ only on indices erased in x is called a* completion *of x.*

▶ **Definition 2.4** (Offline-erasure-resilient tester [9])**.** *For every property $\mathcal{P} \subseteq \Sigma^*$ and parameters $\alpha \in [0, 1)$ and $\varepsilon \in (0, 1)$, an algorithm $\mathcal{T}$ is an* α-offline-erasure-resilient ε-tester *for $\mathcal{P}$ if, given a parameter $n \in \mathbb{N}$ and oracle access to an α-erased input $x \in \Sigma^n$, the algorithm $\mathcal{T}$ accepts with probability at least $\frac{2}{3}$ whenever there exists a completion $y \in \mathcal{P}$ of x and rejects with probability at least $\frac{2}{3}$ whenever every completion y of x is ε-far from $\mathcal{P}$.*

When $\alpha = 0$, the α-offline-erasure-resilient model is the same as the standard model. Another generalization of property testing is tolerant testing, introduced by [25] and extensively studied over the last decades. It can be viewed as guaranteeing resilience to an α fraction of the input being corrupted by an offline adversary.

▶ **Definition 2.5** ((α, ε)-tolerant tester [25])**.** *For every property $\mathcal{P} \subseteq \Sigma^*$ and parameters $\alpha, \varepsilon \in (0, 1)$, an algorithm $\mathcal{T}$ is an* (α, ε)-tolerant tester *for $\mathcal{P}$ if, given a parameter $n \in \mathbb{N}$ and oracle access to input $x \in \Sigma^n$, the algorithm $\mathcal{T}$ accepts with probability at least $\frac{2}{3}$ whenever x is α-close to $\mathcal{P}$ and rejects with probability at least $\frac{2}{3}$ whenever x is ε-far from $\mathcal{P}$.*

When we want to stress comparison to other models we study, we call an (α, ε)-tolerant tester an *α-offline-corruption-resilient ε-tester*.

Online manipulations

In this model, the input string $x \in \Sigma^n$ is accessed via an adversarial oracle $\mathcal{O}$. After answering each query made by the algorithm, the adversary can erase or corrupt a small number of data points. At the beginning of the execution of the algorithm, $\mathcal{O}(i) = x_i$ for all $i \in [n]$. So, the first query is always answered correctly. The number of queries the adversary can manipulate *after* answering each query is parameterized by $t \in \mathbb{N}$. The manipulated values are used by the oracle to answer future queries to the corresponding indices. The algorithm does not know which input locations have been tempered with. As stated in [21], "the actions of the oracle can depend on the input, the queries made so far, and even on the publicly known code that the algorithm is running, but *not* on future coin tosses of the algorithm."

▶ **Definition 2.6** (Adversarial oracle, online testers [21]). *Fix $t \in \mathbb{N}$. A* t-online-erasure *oracle can replace values $\mathcal{O}(i)$ on up to t indices $i \in [n]$ with the erasure symbol $\perp$ after answering each query. A* t-online-corruption *oracle is defined analogously, except that it replaces each symbol with an arbitrary symbol from Σ instead of erasing it. For each $t \in \mathbb{N}$, a* t-online-erasure-resilient ε-tester $\mathcal{T}$ *is defined as in the standard model (Definition 2.2), except that it accesses its input via a t-online-erasure oracle as opposed to querying the input directly. The* t-online-corruption-resilient ε-tester *is defined analogously.*

The standard property testing model is a special case of this enhanced model and corresponds to the case when $t = 0$. Moreover, it is clear from the definition that testing with online erasures is no harder than testing with online corruptions. Specifically, every t-online-corruption-resilient tester can be simulated by a tester that accesses its input via a t-online-erasure oracle – the tester can simply replace each $\perp$ with an arbitrary value from Σ.

3 Query complexity separation

In this section, we prove that testing with an offline adversary can require more queries than testing with an online adversary.

▶ **Theorem 3.1** (Query complexity separation). *Fix $\ell \in \mathbb{N}$ and let $r = r(n) < n$ be a function of the input length n. There exists a property $\mathcal{P} \subseteq \{0,1\}^*$ and a constant $\varepsilon_1 = \varepsilon_1(\ell) \in (0,1)$ such that:*[5]

1. *For all $\varepsilon \in (0,1)$ and $t = (\frac{\varepsilon}{2})^{O(\ell)} \cdot r$, the property $\mathcal{P}$ is t-online-corruption-resiliently ε-testable using $(\frac{2}{\varepsilon})^{O(\ell)}$ queries.*
2. *For all $n \in \mathbb{N}$, $\varepsilon \in (0, \varepsilon_1)$, and $\alpha = \Omega\left(\frac{1}{\log^{(\ell)}(n/r)}\right)$ such that $\varepsilon + \alpha < 1$, every α-offline-erasure-resilient ε-tester for $\mathcal{P}$ must make $\Omega\left(\frac{n}{r \cdot \mathrm{polylog}^{(\ell)}(n/r)}\right)$ queries on inputs of length n.*

For $\ell = 1$ and $r = \frac{n}{\log n}$, we obtain the following corollary.

▶ **Corollary 3.2.** *There exist a property $\mathcal{P}$ such that for all $\varepsilon \in (0,1)$ and $t \leq \mathrm{poly}(\varepsilon) \cdot \frac{n}{\log n}$, the property $\mathcal{P}$ is ε-testable using $\mathrm{poly}(\frac{1}{\varepsilon})$ queries in the t-online-corruption model. However, for all $\alpha = \Omega\left(\frac{1}{\log \log n}\right)$ such that $\alpha + \varepsilon < 1$, every α-offline-erasure-resilient ε-tester requires $\widetilde{\Omega}(\log n)$ queries.*

[5] We use $\log^{(\ell)}$ to denote the log function applied ℓ times.

While Corollary 3.2 shows that testing in the α-offline-erasure model can be harder than testing in the t-online-model for large t and small α, the difference in query complexity is quite mild. To obtain a large gap in query complexity from Theorem 3.1, we fix $\ell \in \mathbb{N}$ and constant proximity parameter $\varepsilon \leq \varepsilon_1(\ell)$, and $r = r(\varepsilon, \ell)$ such that $t = 1$. This yields the following corollary.

▶ **Corollary 3.3.** *Fix $\ell \in \mathbb{N}$ and a constant $\varepsilon = \varepsilon(\ell)$. There exists a property $\mathcal{P}$ such that $\mathcal{P}$ is ε-testable in the 1-online-corruption model using constantly many queries. But, for all $\alpha = \Omega\left(\frac{1}{\log^{(\ell)} n}\right)$ such that $\alpha + \varepsilon < 1$, every α-offline-erasure-resilient ε-tester requires $\Omega(n/\operatorname{polylog}^{(\ell)} n)$ queries.*

One feature of the results in Theorem 3.1 (as well as Corollaries 3.2 and 3.3) is that they hold even when the adversary in the online model is allowed to make corruptions, while the adversary in the offline model is restricted to erasures.

To prove Theorem 3.1, we introduce the following "lifting" result (Lemma 3.5). Informally, the lemma states that every property $\mathcal{P}$ that is testable in the standard model has an encoding $\mathcal{P}'$ that is testable with the same query complexity even in the presence of an online-corruption adversary. This result allows us to transfer existing separations between the standard model and the offline-erasure model to a separation between the online-corruption model and the offline-erasure model.

To "lift" a property $\mathcal{P}$, we simply encoded it with a repetition code, that is, repeat the corresponding input string.

▶ **Definition 3.4** (r-concatenated string x^r and property $\mathcal{P}^r$)**.** *For all $r \in \mathbb{N}$ and strings x, let x^r denote the concatenation of r copies of x, written $x^r[1] \circ x^r[2] \circ \cdots \circ x^r[r]$. Additionally, for a property $\mathcal{P}$ of strings, let $\mathcal{P}^r$ denote the property $\{x^r : x \in \mathcal{P}\}$.*

Our lifting lemma holds for properties of strings over any alphabet.

▶ **Lemma 3.5** (Lifting lemma)**.** *Let Σ be an alphabet and $\delta \in (0,1)$ be a sufficiently small constant. Let $\mathcal{P} \subseteq \Sigma^*$ be a property of strings that is ε-testable in the standard model using $q(m,\varepsilon)$ queries on inputs of length m, where $q(m,\varepsilon) = \Omega\left(\frac{1}{\varepsilon}\right)$. Then, for all $r \in \mathbb{N}$ and $t \leq \frac{\delta \cdot r}{[q(m,\frac{\varepsilon}{2}) \log q(m,\frac{\varepsilon}{2})]^2}$, the property $\mathcal{P}^r$ is t-online-corruption-resiliently ε-testable using $\widetilde{O}(q(m,\frac{\varepsilon}{2}))$ queries on inputs of length $n = m \cdot r$.*

Next we use Lemma 3.5 to prove Theorem 3.1, deferring the proof of Lemma 3.5 to Section 3.1. We leverage the following result of Ben-Eliezer, Fisher, Levi, and Rothblum [5], which separates the offline-erasure-resilient model from the standard model.

▶ **Theorem 3.6** ([5, Theorem 6.2])**.** *For all constant $\ell \in \mathbb{N}$, there exist a property $\mathcal{Q}^{(\ell)} \subseteq \{0,1\}^*$ and a constant $\varepsilon_1 = \varepsilon_1(\ell) \in (0,1)$ such that the following hold:*

1. *For every $\varepsilon \in (0,1)$, the property $\mathcal{Q}^{(\ell)}$ can be ε-tested using $(\frac{2}{\varepsilon})^{O(\ell)}$ queries.*
2. *For all $m \in \mathbb{N}$, $\varepsilon \in (0, \varepsilon_1)$, and $\alpha = \Omega(1/\log^{(\ell)} m)$ satisfying $\varepsilon + \alpha < 1$, every α-erasure resilient ε-tester for $\mathcal{Q}^{(\ell)}$ must make $\Omega(m/(10^\ell \cdot \operatorname{polylog}^{(\ell)} m))$ queries on inputs of length m.*

Proof of Theorem 3.1. Fix $\ell \in \mathbb{N}$ and let $m, r \in \mathbb{N}$ be sufficiently large. Let $\mathcal{P}$ denote the property $\mathcal{Q}^{(\ell)}$ of strings of length m given by Theorem 3.6. By Theorem 3.6, the property $\mathcal{P}$ can be ε-tested using $\left(\frac{2}{\varepsilon}\right)^{O(\ell)}$ queries. Thus, Lemma 3.5 guarantees that for all $t = (\frac{\varepsilon}{2})^{O(\ell)} \cdot r$, the property $\mathcal{P}^r$ is t-online-corruption-resiliently ε-testable using $\left(\frac{2}{\varepsilon}\right)^{O(\ell)}$ queries on inputs of length $n = mr$.

Now we prove that $\mathcal{P}^r$ requires $\Omega\left(\frac{n}{r\,\mathrm{polylog}^{(\ell)}(n/r)}\right)$ queries to ε-test in the α-offline-erasure-resilient model. To do that, we show how to use any α-offline-erasure-resilient ε-tester $\mathcal{T}_r$ for $\mathcal{P}^r$ to construct an α-offline-erasure-resilient ε-tester $\mathcal{T}$ for $\mathcal{P}$ with the same query complexity as $\mathcal{T}_r$. Given a string x^r, let $x^r[i]_j$ denote index j of the i-th copy of x. Observe that for every α-erased input x, the string x^r is also α-erased and the distance of x from $\mathcal{P}$ is the same as the distance of x^r from $\mathcal{P}^r$; moreover, for all $i \in [r]$ and $j \in [m]$, we have $x^r[i]_j = x_j$. Thus, the tester $\mathcal{T}$, given query access to x, can simulate $\mathcal{T}_r$ executed with query access to x^r and output the result given by $\mathcal{T}_r$. By Theorem 3.6, the tester $\mathcal{T}_r$ has query complexity $\Omega\left(\frac{m}{\mathrm{polylog}^{(\ell)}\, m}\right) = \Omega\left(\frac{n}{r\,\mathrm{polylog}^{(\ell)}(n/r)}\right)$ whenever $\alpha = \Omega\left(\frac{1}{\log^{(\ell)}(n/r)}\right)$. ◀

3.1 Proof of the lifting lemma (Lemma 3.5)

Recall that Lemma 3.5 states that if a property $\mathcal{P}$ is testable in the standard model, then the property $\mathcal{P}^r$ is testable in the online-corruption model. Let $\mathcal{T}$ be an ε-tester for $\mathcal{P}$ that has failure probability $\frac{1}{12}$, and query complexity $c_0 \cdot q(m, \varepsilon)$ for some constant $c_0 > 0$ (we can obtain such a tester via standard amplification techniques). We construct a tester $\mathcal{T}'$ for $\mathcal{P}^r$ (Algorithm 1) and show that it is t-online-corruption-resilient. The tester $\mathcal{T}'$ consists of two phases: first, $\mathcal{T}'(s)$ calls REPETITION-TEST (Algorithm 2) to test that s is a repetition of some string, w, and second, $\mathcal{T}'(s)$ simulates $\mathcal{T}$ with query access to w.

For a string s of length $n = r \cdot m$ composed of r substrings each of length m, we use the notation $s[i]_j$ to denote index j, in the i-th substring of s.

Algorithm 1 t-online-corruption-resilient ε-tester $\mathcal{T}'$.

Parameters: length parameter m, repetition parameter r, proximity parameter $\varepsilon \in (0, 1)$
Input: query access to string $s = s[1] \circ \cdots \circ s[r]$ such that $|s[i]| = m$ for each $i \in [r]$
Subroutines: REPETITION-TEST (Algorithm 2) and $c_0 \cdot q(m, \varepsilon)$-query tester $\mathcal{T}$ for $\mathcal{P}$

1: $c_1 \leftarrow 24 \cdot c_0$
2: **if** REPETITION-TEST$(s, \frac{\varepsilon}{2})$ rejects **then reject**
3: simulate $\mathcal{T}$ with proximity parameter $\frac{\varepsilon}{2}$, answering each query j made by $\mathcal{T}$ as follows:
   sample a set of $d = \log(c_1 \cdot q(m, \frac{\varepsilon}{2}))$ uniform indices $i_1, \ldots, i_d \in [r]$
   if there exists a pair of indices $k, k' \in [d]$ such that $s[i_k]_j \neq s[i_{k'}]_j$ **then reject**
   else provide the answer $s[i_1]_j$ to $\mathcal{T}$
4: **if** the simulation of $\mathcal{T}$ rejects **then reject**
5: **else accept**

Algorithm 2 REPETITION-TEST.

Parameters: length parameter m, repetition parameter r, proximity parameter $\varepsilon \in (0, 1)$
Input: query access to string $s = s[1] \circ \cdots \circ s[r]$ such that $|s[i]| = m$ for each $i \in [r]$

1: **repeat** $\frac{2}{\varepsilon}$ times:
   sample $i_1, i_2 \sim [r]$ and $j \sim [m]$ uniformly and independently at random
2: **if** $s[i_1]_j \neq s[i_2]_j$ **then reject**
3: **else accept**

We start by analyzing REPETITION-TEST (Algorithm 2).

▷ Claim 3.7. For all alphabets Σ and parameters $\varepsilon \in (0,1)$ and $r, m \in \mathbb{N}$, REPETITION-TEST is a one-sided error ε-tester for the repetition code defined as $\mathcal{C}_{m,r} = \{s^r : s \in \Sigma^m\}$. It makes $O(\frac{1}{\varepsilon})$ queries and has error probability at most $\frac{1}{6}$.

Proof. Let $s = s[1] \circ \cdots \circ s[r]$, where $s[i] = s[i]_1 \dots s[i]_m$ for all $i \in [r]$, be the input string over the alphabet Σ. If $s \in \mathcal{C}_{m,r}$, then REPETITION-TEST always accepts s.

Suppose that string s is ε-far from $\mathcal{C}_{m,r}$. Let $\text{plurality}_{i\in[r]}(a_i)$ denote a function that takes inputs $a_1, \dots, a_r \in \Sigma$ and outputs the most frequent among them, resolving ties arbitrarily. Let $\hat{w}$ be the string of length m defined by $\hat{w}_j = \text{plurality}_{i\in[r]}\, s[i]_j$ for all $j \in [m]$. Since s is ε-far from $\mathcal{C}_{m,r}$, we can bound the distance between the input string and $\hat{w}$ repeated r times:

$$\delta_H(s, \hat{w}^r) \geq \varepsilon. \tag{1}$$

For all $j \in [m]$ and $a \in \Sigma$, let $p_j(a)$ denote $\Pr_{i\in[r]}[s[i]_j = a]$, the probability that a randomly chosen repetition of the j-th character in s is equal to a. The probability that one iteration of the loop in REPETITION-TEST accepts s is

$$\begin{aligned}
\Pr_{\substack{i_1,i_2\in[r]\\ j\in[m]}}\left[s[i_1]_j = s[i_2]_j\right] &= \frac{1}{m}\sum_{j\in[m]}\sum_{a\in\Sigma}[p_j(a)]^2 && \text{by law of total probability}\\
&\leq \frac{1}{m}\sum_{j\in[m]} p_j(\hat{w}_j)\sum_{a\in\Sigma} p_j(a) && \text{as } p_j(a) \leq p_j(\hat{w}_j)\ \forall j \in [m], a \in \Sigma\\
&= \frac{1}{m}\sum_{j\in[m]} p_j(\hat{w}_j) && \text{since } \sum_{a\in\Sigma} p_j(a) = 1\ \forall j \in [m]\\
&= \Pr_{\substack{i\in[r]\\ j\in[m]}}[s[i]_j = \hat{w}_j] && \text{by law of total probability}\\
&= 1 - \delta_H(s, \hat{w}^r) && \text{by definition of } \delta_H\\
&\leq 1 - \varepsilon && \text{by (1).}
\end{aligned}$$

Hence, each iteration of the loop in REPETITION-TEST rejects with probability at least ε. Therefore, REPETITION-TEST accepts s with probability at most $(1-\varepsilon)^{2/\varepsilon} \leq e^{-\varepsilon\cdot 2/\varepsilon} \leq \frac{1}{6}$. ◁

Next, we analyze the tester $\mathcal{T}'$.

▷ Claim 3.8 ($\mathcal{T}'$ is a tester in the standard model). For all $\varepsilon \in (0,1)$,$r \in \mathbb{N}$, alphabets Σ, and properties $\mathcal{P} \in \Sigma^*$, Algorithm 1 is an ε-tester for $\mathcal{P}^r$ in the standard model (without any adversary). Moreover, it accepts ε-far inputs with probability at most $\frac{1}{6}$, and has one-sided error whenever $\mathcal{T}$ has one-sided error.

Proof. If the input s is in $\mathcal{P}^r$ then $s = w^r$ for some string $w \in \mathcal{P}$. In this case, REPETITION-TEST accepts with probability 1, and $\mathcal{T}'$ accepts with the same probability as $\mathcal{T}$. Next, suppose s is ε-far from $\mathcal{P}^r$. Consider the string $\hat{w} \in \Sigma^m$, where $\hat{w}_j = \text{plurality}_{i\in[r]}\, s[i]_j$ for all $j \in [m]$. Recall that the repetition code is defined as $\mathcal{C}_{m,r} = \{s^r : s \in \Sigma^m\}$. Observe that the distance from s to $\mathcal{C}_{m,r}$ is equal to $\delta_H(s, \hat{w}^r)$. If this distance is at least $\frac{\varepsilon}{2}$ then, by Claim 3.7, REPETITION-TEST accepts with probability at most $\frac{1}{6}$, completing the proof.

Now assume $\delta_H(s, \hat{w}^r) < \frac{\varepsilon}{2}$. We will show that, in this case, the simulation of $\mathcal{T}$ in Algorithm 1 accepts with probability at most $\frac{1}{6}$. Specifically, we consider two failure events: let E_1 be the event that the simulation of $\mathcal{T}$ feeds $\mathcal{T}$ at least one query answer inconsistent with $\hat{w}$, and E_2 be the event that the simulation of $\mathcal{T}$ accepts string $\hat{w}$. We will show that $\Pr[E_1] \leq \frac{1}{12}$ and $\Pr[E_2] \leq \frac{1}{12}$. Then a union bound over E_1 and E_2 completes the proof.

Since s is ε-far from $\mathcal{P}^r$ and $\delta_H(s, \hat{w}^r) < \frac{\varepsilon}{2}$, we get

$$\varepsilon \leq \delta_H(s, \mathcal{P}^r) \leq \delta_H(s, \hat{w}^r) + \delta_H(\hat{w}^r, \mathcal{P}^r) \leq \frac{\varepsilon}{2} + \delta_H(\hat{w}, \mathcal{P}),$$

implying that $\delta_H(\hat{w}, \mathcal{P}) \geq \frac{\varepsilon}{2}$. Since $\mathcal{T}$ is a tester for $\mathcal{P}$ with amplified success probability, and $\hat{w}$ is $\frac{\varepsilon}{2}$-far from $\mathcal{P}$, algorithm $\mathcal{T}$ accepts with probability at most $\frac{1}{12}$ when run on input $\hat{w}$ and with proximity parameter $\frac{\varepsilon}{2}$; that is, $\Pr[E_2] \leq \frac{1}{12}$.

Next we analyze $\Pr[E_1]$. Fix $j \in [m]$, and let a_j denote the answer provided by $\mathcal{T}'$ to query j in the simulation. If $\mathcal{T}'$ does not reject while simulating this query, then

$$\begin{aligned}
\Pr[a_j \neq \hat{w}_j] &= \Pr_{i_1,\ldots,i_d \in [r]}[s[i_k]_j = s[i_{k'}]_j \wedge s[i_k]_j \neq \hat{w}_j \ \forall k, k' \in [d]] \\
&= \Pr_{i_1,\ldots,i_d \in [r]}[s[i_k]_j = s[i_1]_j \ \forall k \in \{2,\ldots,d\} \mid s[i_1]_j \neq \hat{w}_j] \cdot \Pr_{i_1 \in [r]}[s[i_1]_j \neq \hat{w}_j] \\
&\leq 2^{-d+1} \qquad (2) \\
&= 2(c_1 \cdot q(m, \varepsilon/2))^{-1}.
\end{aligned}$$

The inequality in (2) follows from the following simple observation: once $s[i_1]_j$ has been sampled, if it is not the plurality, then for each $k \in \{2,\ldots,d\}$, every subsequent $s[i_k]_j$ is equal to $s[i_1]_j$ with probability at most $\frac{1}{2}$. By our choice of c_1, and a union bound over all the $c_0 \cdot q(m, \varepsilon/2)$ queries made by $\mathcal{T}$, the probability of E_1 is at most $\frac{1}{12}$.

Since $\mathcal{T}'$ can accept only if $E_1 \cup E_2$ occurs, a union bound over E_1 and E_2 implies that $\mathcal{T}'$ is indeed a tester for $\mathcal{P}^r$, and has error probability at most $\frac{1}{6}$ when no adversary is present. ◁

Finally, we prove the lifting lemma.

Proof of Lemma 3.5. First, we argue that the probability $\mathcal{T}'$ (Algorithm 1) queries a corrupted index is small. Recall that $n = m \cdot r$ is the length of the input to $\mathcal{T}'$ and that $c_0 \cdot q(m, \varepsilon)$ is the query complexity of $\mathcal{T}$ with proximity parameter ε. Let $q' = q'(n, \varepsilon) = \frac{8}{\varepsilon} + c_0 \cdot q(m, \frac{\varepsilon}{2}) \log(c_1 \cdot q(m, \frac{\varepsilon}{2}))$ be the number of queries made by $\mathcal{T}'$. Then, there are at most $q't$ corruptions at any point during the execution. Moreover, each query made by $\mathcal{T}'$ is an index that is uniform over the set of corresponding indices in the r different segments of s. It follows that each query made by $\mathcal{T}'$ is corrupted with probability at most $\frac{q't}{r}$. By a union bound over the q' queries,

$$\Pr[\mathcal{T}' \text{ queries a corrupted index}] \leq \frac{(q')^2 t}{r} \leq \frac{1}{6}. \qquad (3)$$

The upper bound follows from the hypothesis that $t \leq \frac{\delta \cdot r}{[q(m,\varepsilon/2) \log q(m,\varepsilon/2)]^2}$, that $q(m, \varepsilon) = \Omega(\frac{1}{\varepsilon})$, and that δ is a sufficiently small constant. By a union bound, the probability that $\mathcal{T}'$ errs in the presence of the adversary is at most the probability $\mathcal{T}'$ errs when no adversary is present plus the probability $\mathcal{T}'$ queries a corrupted index. By Claim 3.8 and (3), this probability is at most $\frac{1}{3}$, which completes the proof of Lemma 3.5. ◀

4 Randomness complexity separation

In this section, we prove our result separating the randomness complexity of testing in the online and offline models.

▶ **Theorem 4.1** (Randomness separation). *For all $\tau = \tau(n)$, there exist a property $\mathcal{P}^{(\tau)} = \bigcup_{n\in\mathbb{N}} \mathcal{P}_n^{(\tau)}$, where $\mathcal{P}_n^{(\tau)} \subseteq [n]^n$, and an algorithm $\mathcal{T}$ that takes $\varepsilon \in (0,1)$ as an input and, for all $\varepsilon \in (0,1)$, is a one-sided error ε-tester for $\mathcal{P}^{(\tau)}$ that make $O(\frac{\tau}{\varepsilon})$ queries. Additionally, the following hold:*

1. *For all $\varepsilon, \alpha \in (0,1)$, tester $\mathcal{T}$ is an α-offline-erasure-resilient ε-tester for $\mathcal{P}^{(\tau)}$. Moreover, $\mathcal{T}$ can be simulated using $O(\log n)$ random bits and no additional queries.*
2. *For all $\varepsilon \in (0,1)$ and $t \leq \left(\frac{0.01\varepsilon\sqrt{n}}{\tau}\right)^2$, tester $\mathcal{T}$ is a t-online-erasure-resilient ε-tester for $\mathcal{P}^{(\tau)}$.*
3. *There exists a constant $\varepsilon > 0$ such that for all $t \in \mathbb{N}$ and $\tau \leq \frac{0.01\sqrt{n}}{\log n}$, every t-online-erasure-resilient ε-tester for $\mathcal{P}^{(\tau)}$ uses $\Omega\left(\frac{\tau \log(t+1)}{\log \tau}\right)$ random bits. Moreover, if the tester has one-sided error, then it uses $\Omega(\tau \log(t+1))$ random bits.*

Theorem 4.1 yields meaningful separations for all settings of τ between $\Omega(\log n)$ and $O\left(\frac{\sqrt{n}}{\log n}\right)$. As an example, the following corollary is derived by setting $\tau = \frac{0.01\varepsilon\sqrt{n}}{\log n}$ and $t = 1$.

▶ **Corollary 4.2.** *There exists a property that can be ε-tested (for every constant ε) using $O\big(\frac{\sqrt{n}}{\log n}\big)$ queries in the presence of either an online or offline adversary. The offline-erasure-resilient tester uses $O(\log n)$ random bits, but there exists a constant ε such that every 1-online-erasure-resilient ε-tester, requires $\Omega\big(\frac{\sqrt{n}}{\operatorname{polylog} n}\big)$ random bits to succeed with constant probability. Moreover, if the 1-online-erasure-resilient tester has one-sided error then it requires $\Omega(\sqrt{n})$ random bits.*

To prove our randomness lower bound, we present a general reduction from randomness-efficient testers in the online-erasure model to query-efficient testers in the standard model.

▶ **Lemma 4.3** (Generalization of Lemma 1.6). *For all $r, t \in \mathbb{N}$ and properties $\mathcal{P}$, if $\mathcal{P}$ is ε-testable in the presence of a t-online-erasure adversary using only r bits of randomness then $\mathcal{P}$ is ε-testable in the standard model using at most $\frac{r}{\log(t+1)}$ queries.*

Proof. Let $\mathcal{T}$ be a t-online-erasure-resilient ε-tester for $\mathcal{P}$ that uses at most r random bits. We construct an adversary that allows $\mathcal{T}$ to make at most $\frac{r}{\log(t+1)}$ non-erased queries. Consider the following random seed elimination adversary $\mathcal{A}$. Given an input x, the description of algorithm $\mathcal{T}$, and query-answer history of $\mathcal{T}$ on input x, the adversary simulates the next query of $\mathcal{T}$ on every random seed $s \in \{0,1\}^r$ that is consistent with the query-answer history of $\mathcal{T}$ thus far and erases the t most queried indices of x. Intuitively, these are the indices that are most likely to be queried by the tester at this step, based on the random seeds that are consistent with the current query-answer history. Each index that the oracle decides not to erase at this step can appear in at most $\frac{1}{t+1}$ fraction of currently relevant random seeds. If $\mathcal{T}$ queries such an index, all random seeds that would have lead to a different query can be eliminated. Thus, each time $\mathcal{T}$ makes a non-erased query, at most a $\frac{1}{t+1}$ fraction of the relevant random seeds remain. Consequently, $\mathcal{T}$ can make at most $\frac{r}{\log(t+1)}$ non-erased queries before $\mathcal{A}$ can exactly determine the random seed being used. Once $\mathcal{A}$ determines the random seed, it can exactly predict the queries of $\mathcal{T}$ and erase all of them.

Next, we use the adversary strategy $\mathcal{A}$ to construct an $\frac{r}{\log(t+1)}$-query tester for $\mathcal{P}$. Let $\mathcal{T}^*$ be defined as follows. Draw a random seed $s^* \in \{0,1\}^r$ and simulate $\mathcal{T}$ with random seed s^*. After making each query, $\mathcal{T}^*$ simulates $\mathcal{A}$ with the query-answer history of $\mathcal{T}$. If $\mathcal{A}$ erases the next query of $\mathcal{T}$, then $\mathcal{T}^*$ answers the query with $\perp$ and continues the simulation (the algorithm $\mathcal{T}^*$ need not make any queries in this case). Otherwise, if $\mathcal{A}$ does not erase

the next query of $\mathcal{T}$, then $\mathcal{T}^*$ queries x, provides $\mathcal{T}$ with the answer, and continues the simulation. Since $\mathcal{T}$ makes at most $\frac{r}{\log(t+1)}$ non-erased queries and successfully tests $\mathcal{P}$, the algorithm $\mathcal{T}^*$ makes at most $\frac{r}{\log(t+1)}$ queries and also successfully tests $\mathcal{P}$. ◀

4.1 The τ-Distinct-Elements property

The property testing problem we use to prove Theorem 4.1 is a version of support size approximation. In support size approximation, one is given sample access to a distribution $\mathcal{D}$ over the domain $[n]$ and asked to approximate $|\text{supp}(\mathcal{D})|$ up to additive error ε. We study the analogous testing problem over strings: Given parameters $\tau \in \mathbb{N}$ and $\varepsilon \in (0,1)$, and a string x of length n, we wish to ε-test whether $|\{x_1, x_2, \ldots, x_n\}| \leq \tau$, that is, to determine whether x has at most τ distinct elements or x is ε-far from every x' (of length n) with at most τ distinct elements.

▶ **Definition 4.4** (τ-Distinct-Elements). *For all $n \in \mathbb{N}$ and $\tau = \tau(n)$, let $\mathcal{P}_n^{(\tau)}$ be the set of strings in $[n]^n$ with at most τ distinct symbols. The property* τ-Distinct-Elements *is defined as $\mathcal{P}^{(\tau)} = \bigcup_{n\in\mathbb{N}} \mathcal{P}_n^{(\tau)}$.*

The following tester for τ-Distinct-Elements appeared in [17]. We include the analysis of its guarantees here for completeness.

▷ Fact 4.5 (A tester for τ-Distinct-Elements). For all $\varepsilon \in (0,1)$, there exists a one-sided error tester for $\mathcal{P}^{(\tau)}$ that uses $O(\frac{\tau}{\varepsilon})$ samples and fails with probability at most $\frac{1}{10}$.

Proof. We first make the following observation: if x is ε-far from $\mathcal{P}^{(\tau)}$, then every collection of τ distinct elements occupies at most a $1-\varepsilon$ fraction of the indices in x. Suppose we sample uniformly random elements from x until we have seen $\tau+1$ distinct elements. Notice that while we have seen at most τ distinct elements, our next sample is a new distinct element with probability at least ε (the at most τ distinct elements we have seen can have probability mass at most $1-\varepsilon$). Let X_i be the number of samples to get the i-th distinct element given we have seen $i-1$ distinct elements, and set $X = \sum_{i=1}^{\tau+1} X_i$. Then $\mathbb{E}[X] \leq (\tau+1)/\varepsilon$, and hence, by Markov's inequality, the probability we need more than $3(\tau+1)/\varepsilon$ samples is at most $\frac{1}{3}$. This analysis inspires the following simple tester:

Sample $s = 3(\tau+1)/\varepsilon$ elements from x and accept if and only if there are at most τ distinct symbols from $[n]$ in the sample (not counting the erasure symbol $\perp$).

Repeating the tester constantly many times suffices to make the failure probability smaller than $\frac{1}{10}$. ◁

Recall that Lemma 4.3 states that to prove a lower bound on the randomness complexity of testing in the online model, it suffices to prove a lower bound on the query complexity of testing in the standard model. In Lemma 4.6, we show a lower bound on the query complexity of testing τ-Distinct-Elements. Our lower bound is a direct corollary of the lower bound of [32].

▶ **Lemma 4.6** (Query lower bound for testing τ-Distinct-Elements). *There exists $\varepsilon \in (0,1)$ such that for all input lengths n, if $\tau \leq \frac{\sqrt{n}}{\log n}$, then the query complexity of ε-testing $\mathcal{P}^{(\tau)}$ is $\Omega(\frac{\tau}{\log \tau})$. Moreover, the query complexity of ε-testing $\mathcal{P}^{(\tau)}$ with one-sided error is $\Omega(\tau)$.*

Proof. We use the following fact to argue that we can assume the tester is sample-based with no loss of generality.

▷ Fact 4.7 (Theorem 6.1 [16]). A property of strings is *symmetric* if it is closed under permutations of the string indices (that is, under reordering of the characters in the string). Let $\mathcal{P}$ be a symmetric property of strings that is ε-testable with query complexity q. Then there exists a sample-based tester for $\mathcal{P}$ with sample complexity $O(q)$.

Since $\mathcal{P}^{(\tau)}$ is symmetric, we can without loss of generality consider sample-based testers. First, we prove the lower bound for two-sided error testers. The essence of our proof is a reduction from the hard instances used in [32] for estimating the support size of a distribution. While we cannot use the lower bound of [32] as a black box, we can nonetheless leverage their hard instances. Indeed, combining Theorem 1, Fact 5, and the remarks regarding the support size of their hard distributions (following Definition 12) from [32], we obtain the following immediate corollary.

▷ Fact 4.8 (Support size approximation lower bound [32]). Let $\alpha > 0$ be a sufficiently small constant. There exist distributions $\mathcal{D}^+$ and $\mathcal{D}^-$ over $[m]$ such that

1. Every element in the support of $\mathcal{D}^+$ or $\mathcal{D}^-$ has probability mass at least $\frac{1}{m}$.
2. $|\mathrm{supp}(\mathcal{D}^+)| \leq m\left(\frac{1}{2}+\alpha\right)$, and $\mathcal{D}^-$ is $\left(\frac{1}{2}-2\alpha\right)$-far from every distribution with support size at most $m\left(\frac{1}{2}+\alpha\right)$.

Moreover, every algorithm that successfully distinguishes $\mathcal{D}^+$ from $\mathcal{D}^-$ with constant probability requires $\Omega\left(\frac{m}{\log m}\right)$ samples.

While the distributions in [32] are defined over $\mathbb{R}$, they all have support size at most m. Thus, we can, without loss of generality, consider distributions over $[m]$. Indeed, given a distribution $\mathcal{D}$ over $\mathbb{R}$, the distribution $\mathcal{D}^\pi$ obtained by choosing a uniformly random permutation π of $[m]$, and then mapping the i-th element in the support of $\mathcal{D}$ to $\pi(i)$, has domain $[m]$ and the same support size as $\mathcal{D}$. Moreover, if two distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ each have support size at most m, then for $b \in \{0,1\}$, the distribution $\mathcal{D}_b^\pi$ can be generated by first, sampling a random permutation π of $[m]$, second, sampling from $\mathcal{D}_b$, and third, sending each distinct sample to the first available element of π. Thus, any algorithm which distinguishes $\mathcal{D}_1^\pi$ and $\mathcal{D}_0^\pi$ can be used to distinguish $\mathcal{D}_0$ and $\mathcal{D}_1$ with the same sample complexity.

To apply Fact 4.8 in our setting, we note that whenever $m \leq \frac{\sqrt{n}}{\log n}$, there exists $x \in [m]^n$ such that the distribution obtained by sampling a uniform index in x has total variation distance at most $\frac{m}{n} = \frac{1}{\sqrt{n}\log n}$ from $\mathcal{D}$. Thus, for sufficiently large n, there exist strings $x^+, x^- \in [m]^n$ such that x^+ has at most $m(\frac{1}{2}+\alpha)$ distinct symbols, and the distributions generated by sampling from x^+ and x^- have total variation distance $\frac{1}{\sqrt{n}\log n}$ from $\mathcal{D}^+$ and $\mathcal{D}^-$, respectively. Let $\tau = m(\frac{1}{2}+\alpha)$ and $\varepsilon = \frac{1}{2} - 2\alpha - o(1)$, then $x^+ \in \mathcal{P}^{(\tau)}$ and x^- is ε-far from $\mathcal{P}^{(\tau)}$. By Fact 4.8, every algorithm for distinguishing x^+ from x^- uses $\Omega\left(\frac{m}{\log m}\right) = \Omega\left(\frac{\tau}{\log \tau}\right)$ samples. By Fact 4.7, every tester for $\mathcal{P}^{(\tau)}$ has query complexity $\Omega\left(\frac{\tau}{\log \tau}\right)$.

To prove the one-sided error lower bound, it suffices to note that every one-sided error algorithm must accept if its sample contains at most τ symbols. Thus, every one-sided error tester requires $\Omega(\tau)$ samples. This completes the proof of Lemma 4.6. ◀

4.2 Proof of the randomness separation (Theorem 4.1)

To complete the proof of the theorem, we argue that the tester given by Fact 4.5 is a valid tester for τ-Distinct-Elements in both the online and offline erasure models. To prove the lower bound on the randomness complexity of testing τ-Distinct-Elements in the online model, we combine the query lower bound of Lemma 4.6 with the reduction stated in Lemma 4.3. To

prove the upper bound on the randomness complexity of testing τ-Distinct-Elements in the offline model, we argue that the offline tester can be simulated with few bits of randomness at a small cost in error probability.

We first prove the part of the theorem regarding testing τ-Distinct-Elements in the offline erasure model.

Proof of Theorem 4.1, Item 1. First, for all $\tau \in \mathbb{N}$, we construct an offline-erasure-resilient tester for τ-Distinct-Elements, and second, we argue that the tester can be simulated with $O(\log n)$ random bits and no additional queries. Let $\mathcal{T}$ be the tester given by Fact 4.5. Recall that $\mathcal{T}$ accepts if and only if the number of nonerased symbols in the sample is at most τ. Below, we argue that $\mathcal{T}$ is an α-offline-erasure-resilient ε-tester.

Let x be an α-erased string in $[n]^n$. If some completion of x is in $\mathcal{P}^{(\tau)}$, then the number of nonerased symbols in the sample drawn by $\mathcal{T}$ is at most τ, so the algorithm accepts. Next we argue that $\mathcal{T}$ accepts with probability at most $\frac{1}{4}$ when every completion of x is ε-far from $\mathcal{P}^{(\tau)}$. Let $x^{\perp}$ denote the erased portion of x, and $x^{\top}$ denote the nonerased portion of x. Notice that $|x^{\top}| = (1-\alpha)n$, and thus if $x^{\top}$ is $\frac{\varepsilon}{1-\alpha}$-close to $\mathcal{P}^{(\tau)}$, then there is a completion of x that is ε-close to $\mathcal{P}^{(\tau)}$ (change εn symbols in $x^{\top}$ and fill out $x^{\perp}$ with an arbitrary symbol from $x^{\top}$). Thus, if every completion of x is ε-far from $\mathcal{P}^{(\tau)}$, then $x^{\top}$ is $\frac{\varepsilon}{1-\alpha}$-far from $\mathcal{P}^{(\tau)}$. By the proof of Fact 4.5, a set of $O\left(\frac{\tau(1-\alpha)}{\varepsilon}\right)$ uniform and independent samples from $x^{\top}$ contains at least $\tau+1$ distinct symbols with probability at least $\frac{9}{10}$. Moreover, a sample of $O\left(\frac{\tau}{\varepsilon}\right)$ elements from x contains $\Omega\left(\frac{\tau(1-\alpha)}{\varepsilon}\right)$ such samples from $x^{\top}$ with probability at least $\frac{9}{10}$. By the union bound, $\mathcal{T}$ accepts x with probability at most $\frac{1}{4}$.

It remains to show that $\mathcal{T}$ can be simulated by a tester $\mathcal{T}'$ that uses $O(\log n)$ random bits and has the same query complexity as $\mathcal{T}$. Fact 4.9 states that any randomized oracle machine that solves a promise problem can be simulated using $\log n + \log\log|\Sigma| + O(1)$ random bits, where $|\Sigma|$ is the size of the alphabet. This fact was originally proven by Goldreich and Sheffet [18] and is stated as an exercise in [14]. We use the statement from the exercise since it is more convenient for our setting.

▷ Fact 4.9 (Exercise 1.21 [14]). Fix $n \in \mathbb{N}$ and let Π be a promise problem regarding strings in Σ^n. Suppose that $\mathcal{M}$ is a randomized q-query oracle machine that solves Π with error probability at most $\frac{1}{4}$. Then there exists a randomized q-query oracle machine $\mathcal{M}'$ that solves Π with error probability at most $\frac{1}{3}$ and $\log n + \log\log|\Sigma| + O(1)$ random bits.

Let Π be the promise problem defined by α-offline-erasure-resilient ε-testing $\mathcal{P}^{(\tau)}$. By the first part of the proof, there exists a randomized oracle machine $\mathcal{T}$ that solves Π with error probability at most $\frac{1}{4}$. Applying Fact 4.9 directly, we see that $\mathcal{T}$ can be simulated by a tester $\mathcal{T}'$ using $\log n + \log\log|\Sigma| + O(1)$ random bits; since $\Sigma = [n]$, tester $\mathcal{T}'$ uses $O(\log n)$ random bits, and has error probability at most $\frac{1}{3}$. To complete the proof, it suffices to note that $\mathcal{T}'$ has the same query complexity as $\mathcal{T}$. ◀

Next, we prove the part of the theorem regarding testing τ-Distinct-Elements in the online erasure model.

Proof of Theorem 4.1, Items 2 and 3. We will show that for all $\tau \in \mathbb{N}$, the tester given by Fact 4.5 is online-erasure-resilient. Let $\mathcal{T}$ be the tester for τ-Distinct-Elements given by Fact 4.5. Recall that $\mathcal{T}$ accepts if and only if the number of nonerased symbols in its sample is at most τ. We show that $\mathcal{T}$ can test in the presence of a t-online-erasure-oracle, by arguing that with high constant probability, none of the queries made by $\mathcal{T}$ are erased. Notice that with proximity parameter ε, the tester makes $O\left(\frac{\tau}{\varepsilon}\right)$ queries. Thus, there are

$O\left(\frac{t\cdot\tau}{\varepsilon}\right)$ erasures at every point during the execution of $\mathcal{T}$. Since $\mathcal{T}$ makes uniform and independent queries, the probability that any particular query is erased is $O\left(\frac{t\cdot\tau}{n\varepsilon}\right)$. Since $\mathcal{T}$ makes $O\left(\frac{\tau}{\varepsilon}\right)$ queries, we can apply the union bound to see that some query is erased with probability at most $O\left(\frac{t\cdot\tau^2}{n\varepsilon^2}\right)$. By the hypothesis that $t \leq \left(\frac{.01\varepsilon\sqrt{n}}{\tau}\right)^2$, we have $O\left(\frac{t\cdot\tau^2}{n\varepsilon^2}\right) \leq .01$ – that is, $\mathcal{T}$ sees an erasure with probability at most .01. By Fact 4.5, the algorithm $\mathcal{T}$ is a t-online-erasure-resilient ε-tester for $\mathcal{P}^{(\tau)}$. Moreover, since an erasure cannot increase number of distinct nonerased symbols in x, tester $\mathcal{T}$ has one-sided error. This completes the proof of Item 2.

To see why Item 3 holds, recall that by Lemma 4.6, there exists $\varepsilon \in (0,1)$ such that for all $n, \tau \in \mathbb{N}$ with $\tau \leq \frac{\sqrt{n}}{\log n}$, every ε-tester for $\mathcal{P}^{(\tau)}$ has query complexity $\Omega\left(\frac{\tau}{\log \tau}\right)$. Similarly every tester with one-sided error has query complexity $\Omega(\tau)$. Thus, applying Lemma 4.3, the reduction from query complexity to randomness complexity, suffices to complete the proof. ◀

References

1 Tomer Adar and Eldar Fischer. Refining the adaptivity notion in the huge object model. In Amit Kumar and Noga Ron-Zewi, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2024, August 28-30, 2024, London School of Economics, London, UK*, volume 317 of *LIPIcs*, pages 45:1–45:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.APPROX/RANDOM.2024.45.

2 Tomer Adar, Eldar Fischer, and Amit Levi. Support testing in the huge object model. In Amit Kumar and Noga Ron-Zewi, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2024, August 28-30, 2024, London School of Economics, London, UK*, volume 317 of *LIPIcs*, pages 46:1–46:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.APPROX/RANDOM.2024.46.

3 Vipul Arora, Esty Kelman, and Uri Meir. On optimal testing of linearity. In *Symposium on Simplicity in Algorithms (SOSA)*. SIAM, 2025. To appear.

4 Aleksandrs Belovs. Adaptive lower bound for testing monotonicity on the line. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pages 31:1–31:10, 2018. doi:10.4230/LIPIcs.APPROX-RANDOM.2018.31.

5 Omri Ben-Eliezer, Eldar Fischer, Amit Levi, and Ron D. Rothblum. Hard properties with (very) short PCPPs and their applications. In *Proceedings, Innovations in Theoretical Computer Science (ITCS)*, pages 9:1–9:27, 2020. doi:10.4230/LIPIcs.ITCS.2020.9.

6 Omri Ben-Eliezer, Esty Kelman, Uri Meir, and Sofya Raskhodnikova. Property testing with online adversaries. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference, ITCS 2024, January 30 to February 2, 2024, Berkeley, CA, USA*, volume 287 of *LIPIcs*, pages 11:1–11:25. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.ITCS.2024.11.

7 Arnab Bhattacharyya, Elena Grigorescu, Kyomin Jung, Sofya Raskhodnikova, and David P. Woodruff. Transitive-closure spanners. *SIAM Journal on Computing*, 41(6):1380–1425, 2012. doi:10.1137/110826655.

8 Deeparnab Chakrabarty and C. Seshadhri. An optimal lower bound for monotonicity testing over hypergrids. *Theory of Computing*, 10:453–464, 2014. doi:10.4086/toc.2014.v010a017.

9 Kashyap Dixit, Sofya Raskhodnikova, Abhradeep Thakurta, and Nithin Varma. Erasure-resilient property testing. *SIAM Journal on Computing*, 47(2):295–329, 2018. doi:10.1137/16M1075661.

10 Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pages 97–108, 1999. doi:10.1007/978-3-540-48413-4_10.

11 Funda Ergün, Sampath Kannan, Ravi Kumar, Ronitt Rubinfeld, and Mahesh Viswanathan. Spot-checkers. *J. Comput. Syst. Sci.*, 60(3):717–751, 2000. doi:10.1006/JCSS.1999.1692.

12 Eldar Fischer. On the strength of comparisons in property testing. *Inf. Comput.*, 189(1):107–116, 2004. doi:10.1016/j.ic.2003.09.003.

13 Eldar Fischer and Lance Fortnow. Tolerant versus intolerant testing for boolean properties. *Theory Comput.*, 2(9):173–183, 2006. doi:10.4086/TOC.2006.V002A009.

14 Oded Goldreich. *Introduction to property testing*. Cambridge University Press, 2017. doi:10.1017/9781108135252.

15 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998. doi:10.1145/285055.285060.

16 Oded Goldreich and Dana Ron. On sample-based testers. In Tim Roughgarden, editor, *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 337–345. ACM, 2015. doi:10.1145/2688073.2688080.

17 Oded Goldreich and Dana Ron. Testing distributions of huge objects. *CoRR*, abs/2212.12802, 2022. doi:10.48550/arXiv.2212.12802.

18 Oded Goldreich and Or Sheffet. On the randomness complexity of property testing. *Computational Complexity*, 19:99–133, 2010. doi:10.1007/S00037-009-0282-4.

19 Madhav Jha and Sofya Raskhodnikova. Testing and reconstruction of Lipschitz functions with applications to data privacy. *SIAM Journal on Computing*, 42(2):700–731, 2013. doi:10.1137/110840741.

20 Renato Ferreira Pinto Jr. and Nathaniel Harms. Testing support size more efficiently than learning histograms. *CoRR*, 2024. arXiv:2410.18915, doi:10.48550/arXiv.2410.18915.

21 Iden Kalemaj, Sofya Raskhodnikova, and Nithin Varma. Sublinear-time computation in the presence of online erasures. *Theory Comput.*, 19 (1):1–48, 2023. URL: https://theoryofcomputing.org/articles/v019a001/v019a001.pdf, doi:10.4086/TOC.2023.V019A001.

22 Esty Kelman, Ephraim Linder, and Sofya Raskhodnikova. Online versus offline adversaries in property testing. *CoRR*, 2024. arXiv:2411.18617.

23 Amit Levi, Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and Nithin Varma. Erasure-resilient sublinear-time graph algorithms. *ACM Trans. Comput. Theory*, 14(1):1:1–1:22, 2022. doi:10.1145/3488250.

24 Dor Minzer and Kai Zhe Zheng. Adversarial low degree testing. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 4395–4409. SIAM, 2024. doi:10.1137/1.9781611977912.154.

25 Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *J. Comput. Syst. Sci.*, 72(6):1012–1042, 2006. doi:10.1016/j.jcss.2006.03.002.

26 Sofya Raskhodnikova. Monotonicity testing. *Masters Thesis, MIT*, 1999.

27 Sofya Raskhodnikova. Testing if an array is sorted. *Encyclopedia of Algorithms*, pages 2219–2222, 2016. doi:10.1007/978-1-4939-2864-4_700.

28 Sofya Raskhodnikova, Dana Ron, Amir Shpilka, and Adam Smith. Strong lower bounds for approximating distribution support size and the distinct elements problem. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 559–569. IEEE Computer Society, 2007. doi:10.1109/FOCS.2007.67.

29 Sofya Raskhodnikova, Noga Ron-Zewi, and Nithin Varma. Erasures versus errors in local decoding and property testing. *Random Structures and Algorithms*, 59(4):640–670, 2021. doi:10.1002/rsa.21031.

30 Sofya Raskhodnikova and Nithin Varma. Brief announcement: Erasure-resilience versus tolerance to errors. In *Proceedings, International Colloquium on Automata, Languages and Programming (ICALP)*, pages 111:1–111:3, 2018. doi:10.4230/LIPIcs.ICALP.2018.111.

31 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996. doi:10.1137/S0097539793255151.

32 Gregory Valiant and Paul Valiant. A CLT and tight lower bounds for estimating entropy. *Electron. Colloquium Comput. Complex.*, TR10-179, 2010. URL: https://eccc.weizmann.ac.il/report/2010/179.

33 Alek Westover, Edward Yu, and Kai Zheng. New direct sum tests. *CoRR*, abs/2409.10464, 2024. doi:10.48550/arXiv.2409.10464.

Graph Reconstruction via MIS Queries



Christian Konrad
School of Computer Science, University of Bristol, UK

Conor O'Sullivan
School of Computer Science, University of Bristol, UK

Victor Traistaru
School of Computer Science, University of Bristol, UK

Abstract

In the Graph Reconstruction (GR) problem, a player initially only knows the vertex set V of an input graph $G = (V, E)$ and is required to learn its set of edges E. To this end, the player submits queries to an oracle and must deduce E from the oracle's answers.

Angluin and Chen [Journal of Computer and System Sciences, 2008] resolved the number of Independent Set (IS) queries necessary and sufficient for GR on m-edge graphs. In this setting, each query consists of a subset of vertices $U \subseteq V$, and the oracle responds with a boolean, indicating whether U is an independent set in G. They gave algorithms that use $O(m \cdot \log n)$ IS queries, which is best possible.

In this paper, we initiate the study of GR via Maximal Independent Set (MIS) queries, a more powerful variant of IS queries. Given a query $U \subseteq V$, the oracle responds with any, potentially adversarially chosen, maximal independent set $I \subseteq U$ in the induced subgraph $G[U]$.

We show that, for GR, MIS queries are strictly more powerful than IS queries when parametrized by the maximum degree Δ of the input graph. We give tight (up to poly-logarithmic factors) upper and lower bounds for this problem:

1. We observe that the simple strategy of taking uniform independent random samples of V and submitting those to the oracle yields a non-adaptive randomized algorithm that executes $O(\Delta^2 \cdot \log n)$ queries and succeeds with high probability. This should be contrasted with the fact that $\Omega(\Delta \cdot n \cdot \log(\frac{n}{\Delta}))$ IS queries are required for such graphs, which shows that MIS queries are strictly more powerful than IS queries. Interestingly, combining the strategy of taking uniform random samples of V with the probabilistic method, we show the existence of a deterministic non-adaptive algorithm that executes $O(\Delta^3 \cdot \log(\frac{n}{\Delta}))$ queries.
2. Regarding lower bounds, we prove that the additional Δ factor when going from randomized non-adaptive algorithms to deterministic non-adaptive algorithms is necessary. We show that every non-adaptive deterministic algorithm requires $\Omega(\Delta^3/\log^2 \Delta)$ queries. For arbitrary randomized adaptive algorithms, we show that $\Omega(\Delta^2)$ queries are necessary in graphs of maximum degree Δ, and that $\Omega(\log n)$ queries are necessary, even when the input graph is an n-vertex cycle.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Query Complexity, Graph Reconstruction, Maximal Independent Set Queries

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.66

1 Introduction

Query algorithms for graph problems have recently received significant attention. In this setting, algorithms are granted access to the input graph solely via a (usually easy-to-compute) subroutine, which is referred to as the oracle, and the complexity of an algorithm is measured by the number of subroutine/oracle calls.

In the literature, a large number of different query models have been considered. Queries can either be *local* or *global*, depending on whether they reveal only local information, e.g., vertex degree queries [13] or queries that, for any i, reveal the ith neighbour of a vertex [17], or global information, e.g., (bipartite) independent set queries [7, 10, 1, 3] or maximal


© Christian Konrad, Conor O'Sullivan, and Victor Traistaru;
licensed under Creative Commons License CC-BY 4.0

matching queries [12, 20]. Depending on the application, it is generally desirable to obtain *non-adaptive* query algorithms, i.e., algorithms where the different queries do not depend on each other's outcomes, since such queries can be executed simultaneously and therefore admit straightforward parallel implementations. We also say that a query algorithm requires k *rounds of adaptivity*, for some integer k, if the queries executed by the algorithm can be partitioned into k groups such that the queries in the ith group only depend on the outcomes of the queries in groups $1, \ldots, i-1$.

Graph Reconstruction. In this work, we consider query algorithms for the Graph Reconstruction (GR) problem. In GR, a player initially only knows the vertex set V of a graph $G = (V, E)$ and is tasked with learning the edge set E, by submitting a sequence of adaptive or non-adaptive queries to the oracle, and by deducing the edge set E from the oracle's answers.

Query algorithms for GR have been extensively studied under various query models, including Independent Set (IS) queries [18, 11, 6, 5, 7, 1], Distance queries [24, 19, 21, 25], Betweenness queries [2, 26], and in the setting where an algorithm receives random vertex-induced subgraphs or submatrices of the adjacency matrix [22]. Very relevant to our work are IS queries, where a query consists of a subset of vertices $U \subseteq V$, and the oracle returns a boolean, indicating whether U is an independent set in G. IS queries for GR were originally studied for recovering simple graphs, such as matchings and Hamiltonian cycles [18, 11, 6], since these settings have direct applications to genome sequencing. Angluin and Chen [7] were the first to consider general graphs and showed that m-edge graphs can be reconstructed with $O(m \cdot \log n)$ IS queries via an adaptive deterministic algorithm or a randomized algorithm with limited adaptivity (see also [1] who reduced the number of adaptive rounds), and this is also best possible.

MIS Queries. In this work, we initiate the study of GR under Maximal Independent Set (MIS) queries. In this setting, similar to IS queries, a query also consists of a subset of vertices $U \subseteq V$. The oracle, however, responds with any, potentially adversarially chosen, maximal independent set $I \subseteq U$ in the subgraph induced by the queried vertices $G[U]$.

MIS queries are global queries and are similar in spirit to the Maximal Matching queries considered in [12, 20], where the oracle returns a maximal matching in the subgraph induced by the query vertices [12], or in the subgraph spanned by the queried edges that are also contained in the input graph [20]. They are at least as powerful as IS queries since an IS query $U \subseteq V$ can be answered by an MIS oracle by exploiting the connection:

$$\text{IS-Query}(U) = \texttt{true} \iff \text{MIS-Query}(U) = U \ .$$

We observe that computing an MIS is a simple task that can be solved in linear time by a Greedy algorithm and even in sublinear time on graphs of bounded-neighborhood independence [9]. The MIS problem can also be solved efficiently in many restricted computational models, e.g., an MIS can be computed in $O(\log\log\Delta)$ rounds in the Congested-Clique model [16], in $O(\log\Delta)$ rounds in the LOCAL model of distributed computing [15, 27], and in $O(\log\log n)$ passes in the semi-streaming model [4, 8]. This task can therefore be regarded as a building block of more elaborate algorithms and may be outsourced to an oracle that maintains an efficient implementation.

Our key motivation is to answer the question as to whether MIS queries are strictly more powerful than IS queries, i.e., whether enhancing the oracle answer by providing a maximal independent set rather than just indicating whether the query set is independent yields significant savings in the number of queries required.

Table 1 Overview of our results. Our results are parametrized by the maximum degree Δ of the input graph. Both our algorithms are non-adaptive and require knowledge of Δ in advance. We also give adaptive counterparts with similar query complexity that do not require advanced knowledge of Δ. The lower bounds for randomized algorithms hold for adaptive algorithms. The lower bound for deterministic algorithms holds for non-adaptive algorithms.

	Algorithm	Lower Bound
Randomized	$O(\Delta^2 \log n)$ (**Theorem 1**)	$\Omega(\Delta^2 + \log n)$ (**Theorems 13 and 14**)
Deterministic	$O(\Delta^3 \log(\frac{n}{\Delta}))$ (**Corollary 7**)	$\Omega(\Delta^3/\log^2 \Delta)$ (**Corollary 12**)

1.1 Our Results

In this paper, we show that MIS queries can be strictly more powerful than IS queries, but this is also not always the case.

As previously mentioned, Angluin and Chen showed that $O(m \log n)$ IS queries are sufficient for reconstructing a graph on m edges. One of our lower bound results (**Theorem 14**) implies that there are m-edge graphs that require $\Omega(m)$ MIS queries to be reconstructed, which show that, up to a logarithmic factor, MIS queries are not more powerful than IS queries on the class of m-edge graphs. However, when considering the class of graphs with maximum degree Δ, for some integer Δ, significant savings can be achieved:

Algorithms. We give a randomized algorithm for reconstructing an n-vertex graph with maximum degree Δ that uses $O(\Delta^2 \log n)$ non-adaptive queries and succeeds with high probability[1] (**Theorem 1**). This result shows that, for the class of graphs of maximum degree Δ, MIS queries are stronger than IS queries since an information-theoretic lower bound similar to the one given in [7] shows that $\Omega(\Delta n \log(\frac{n}{\Delta}))$ IS queries are needed to reconstruct a graph with maximum degree Δ (**Corollary 16** in Appendix A). Observe that, for the class of constant degree graphs, $O(\log n)$ MIS queries are sufficient, but $\Omega(n \log n)$ IS queries are necessary.

We also investigate whether randomization is necessary to obtain algorithms with low query complexity. Using the probabilistic method, we show that there exists a non-adaptive deterministic query algorithm that executes $O(\Delta^3 \log(\frac{n}{\Delta}))$ queries (**Corollary 7**).

Our non-adaptive randomized and deterministic algorithms assume that the maximum degree Δ is known in advance. This cannot be avoided since, as proved in Theorem 14, any algorithm that executes $o(n^2)$ queries cannot solve all instances with $\Delta = \Theta(n)$. We show that the non-adaptive algorithms above can be turned into adaptive algorithms with query complexities that are by at most a constant factor larger. These adaptive algorithms do not require any information about Δ in order to operate and only require $O(\log \Delta)$ rounds of adaptivity (**Corollary 9**).

Lower Bounds. We give lower bounds on the number of queries required by both non-adaptive deterministic algorithms and adaptive randomized algorithms.

First, we show that every non-adaptive deterministic query algorithm requires $\Omega(\Delta^3/\log^2 \Delta)$ queries (**Corollary 12**), which renders our deterministic algorithm optimal, up to poly-logarithmic factors. This result together with our non-adaptive randomized

[1] As it is standard, we say that an event related to the input graph $G = (V, E)$ occurs *with high probability* if the probability is $1 - \frac{1}{\text{poly } n}$, where $n = |V|$.

algorithm (**Theorem 1**) establishes a separation result between non-adaptive randomized and non-adaptive deterministic algorithms since our randomized algorithm only requires $O(\Delta^2 \log n)$ queries.

Next, we show that every adaptive randomized query algorithm requires $\Omega(\log n)$ queries, even if the input graph is guaranteed to be an n-vertex cycle (**Theorem 13**). Furthermore, we show that every adaptive randomized query algorithm requires $\Omega(\Delta^2)$ queries, for any Δ (**Theorem 14**). These lower bounds show that the number of queries executed by our randomized algorithm is optimal, up to a logarithmic factor, and that the logarithmic dependency on n cannot be avoided entirely.

For an overview of our results, see Table 1.

1.2 Techniques

Algorithms. The starting point for all our algorithms is an algorithm by Angluin and Chen [7] for GR that uses IS queries. Our randomized algorithm is in fact identical to their algorithm, up to the use of a different sampling probability. It works as follows:

The key idea is to learn all the non-edges rather than all the edges of the input graph. To this end, we sample sufficiently many random vertex-induced subgraphs $G[V_i] \subseteq G$, for integers i, such that, for every non-edge $uv \in (V \times V) \setminus E(G)$, the probability that both u and v are contained in V_i and are isolated in $G[V_i]$ is $\Theta(\frac{1}{\Delta^2})$. This is achieved by including every vertex in V_i with probability $\frac{1}{\Delta+1}$. Observe that when u and v are isolated in $G[V_i]$ then the oracle necessarily needs to include both u and v in the returned maximal independent set. The returned maximal independent set therefore serves as a witness that proves that the potential edge uv is not contained in the input graph. We then argue that, after repeating this process $O(\Delta^2 \log n)$ times, we have learnt all non-edges with high probability, which allows us to identify the edges of the input graph by complementing the set of non-edges.

Angluin and Chen use this idea in the context of IS queries for graphs with maximum degree Δ, where the significantly lower inclusion probability of each vertex into each sample V_i of $\frac{1}{\sqrt{\Delta n}}$ needs to be used.

Our deterministic algorithm is built on a similar idea. We show that, when taking $\ell = \Theta(\Delta^3 \log(\frac{n}{\Delta}))$ random subsets $V_1, \ldots, V_\ell \subseteq V$ as above, i.e., by inserting every vertex $v \in V$ into any subset V_i with probability $\frac{1}{\Delta+1}$, then, for any tuple $(\{u, v\}, \{w_1, \ldots, w_{2\Delta}\})$ of $2\Delta + 2$ disjoint vertices, which we also refer to as a *witness*, there exists a set V_i such that $u, v \in V_i$ but $w_1, \ldots, w_{2\Delta} \notin V_i$ with positive probability. Since this event happens with non-zero probability, such a family of subsets $V_1, \ldots, V_\ell$ exists. Our deterministic algorithm then queries all of these subsets $(V_i)_{1 \leq i \leq \ell}$. We now claim that our algorithm learns every non-edge $uv \in (V \times V) \setminus E$: We know that there exists a set V_i such that $u, v \in V_i$ but $(\Gamma(v) \cup \Gamma(u)) \cap V_i = \varnothing$ since $|\Gamma(v) \cup \Gamma(u)| \leq 2\Delta$ holds. The vertices u and v are thus isolated in $G[V_i]$ and therefore necessarily reported in the oracle answer, which provides a proof to the algorithm that the edge uv is not contained in the input graph.

Both our non-adaptive randomized and deterministic algorithms require advanced knowledge of Δ, which, as previously argued, is unavoidable. We turn both of these algorithms into adaptive algorithms that require only $O(\log \Delta)$ rounds of adaptivity and whose total number of queries executed is only by a constant factor larger. This is achieved by successively running our algorithms for the guesses $D = 1, 2, 4, 8, \ldots$ for the maximum degree Δ until a final guess $\Delta < D \leq 2\Delta$ is used, in which case it is easy to see that the graph is correctly reconstructed. While this is a standard doubling argument, the non-trivial part of the argument is to identify when the condition $\Delta < D \leq 2\Delta$ is reached since Δ is now unknown. Denoting by F the set of non-edges learnt by the execution of one of our algorithms when

invoked on the current guess D, we show that if the maximum degree in the graph spanned by the edges $(V \times V) \setminus F$ is at least D then we have indeed learnt all the non-edge of the input graph, and thus also reconstructed the graph.

Lower Bounds. We first discuss our $\Omega(\Delta^3/\log^2 \Delta)$ lower bound for deterministic non-adaptive algorithms. Witnesses, i.e., tuples $(\{u,v\},\{w_1,\ldots,w_{2\Delta}\})$ of $2\Delta+2$ disjoint vertices, play a key role in our lower bound argument. We argue that any deterministic non-adaptive query algorithm must be such that, for every witness $(\{u,v\},\{w_1,\ldots,w_{2\Delta}\})$ of disjoint $2\Delta+2$ vertices, there exists a query Q_i such that $u,v \in Q_i$ and $\{w_1,\ldots,w_{2\Delta}\} \cap Q_i = \varnothing$. We call a set of ℓ queries that fulfills this property a Δ-*Query-Scheme of size* ℓ. To see that the previous property is true, for the sake of a contradiction, suppose that this is not the case and there exists a witness $(\{u,v\},\{w_1,\ldots,w_{2\Delta}\})$ that does not fulfill these properties, i.e., whenever u,v is included in a query then at least one of the vertices $w_1,\ldots,w_{2\Delta}$ is included in this query as well. We claim that the two graphs G_1 and G_2, where, in both graphs, $w_1,\ldots,w_\Delta$ are incident on u and $w_{\Delta+1},\ldots,w_{2\Delta}$ are incident on v, and uv is an edge in G_1 but not in G_2 cannot be distinguished[2]. Indeed, we claim that, for any query executed, the oracle can always report an independent set that contains at most one of the two vertices u,v, even on graph G_2. This is because whenever u,v is included in a query then at least one of the vertices $(w_i)_{1\le i\le 2\Delta}$ is also included in this query. The oracle can therefore include w_i in the oracle answer, which implies that at most one of u and v will also be included. Both graphs G_1 and G_2 are therefore consistent with all oracle answers and are thus indistinguishable, a contradiction.

Our task, therefore, is to prove that any Δ-Query-Scheme must be of size at least $\Omega(\Delta^3/\log^2 \Delta)$. We achieve this by combining two separate arguments that address small queries, i.e., queries $V_i \subseteq V$ of size at most $|V_i| \le t := C \cdot \frac{n \ln(\Delta)}{\Delta}$ and large queries of size larger than t, respectively: For any pair of vertices $\{u,v\}$, small queries are such that they cover many witnesses $(\{u,v\},\{w_1,\ldots,w_{2\Delta}\})$, for many different vertices $(w_i)_{1\le i\le 2\Delta}$, since the vertices $(w_i)_{1\le i\le 2\Delta}$ are chosen from $V \setminus V_i$, however, on average, vertex pairs $\{u,v\}$ cannot be included in many small queries (at most $\binom{t}{2}$). In contrast, vertex pairs $\{u,v\}$ can be included in many large queries, however, those queries do not cover many witnesses $(\{u,v\},\{w_1,\ldots,w_{2\Delta}\})$, for vertices $(w_i)_{1\le i\le 2\Delta}$. We obtain our result by combining these observations, see the proof of Lemma 10.

Next, we discuss our $\Omega(\log n)$ queries lower bound for adaptive randomized algorithms on n-vertex cycles. Observe that, in an n-vertex cycle $G=(V,E)$, every pair of vertices $u,v \in V$ is such that $\Gamma(u) \neq \Gamma(v)$. Hence, for a query algorithm to be able to distinguish any two vertices of the input graph, the algorithm is required to obtain oracle answer maximal independent sets such that u and v do not behave the same in all returned independent sets. We show that $\Omega(\log n)$ queries are needed to distinguish between all the vertices. Our argument is based on the observation that, given a set of vertices $U \subseteq V$ that are so far indistinguishable and a query set $V_i \subseteq V$, the returned maximal independent set I_i only allows us to differentiate between the vertices of U that are not queried, queried and contained in I_i, and queried and not contained in I_i. Every query thus allows us to partition any set of vertices that is still indistinguishable into only three subsets, which implies that $\Omega(\log_3 n) = \Omega(\log n)$ queries are needed to distinguish between all the vertices.

[2] This construction generates a maximum degree of $\Delta+1$ in G_1, which is technically not allowed since we consider algorithms that run on graphs of maximum degree Δ. To circumvent this issue, in our actual proof we therefore relate algorithms that operate on maximum degree Δ graphs to $(\Delta-1)$-Query-Schemes.

Last, we discuss our $\Omega(\Delta^2)$ queries lower bound for arbitrary randomized adaptive algorithms. We work with the family of graphs $\mathcal{G}$ obtained from the family of all balanced bipartite graphs $G = (A, B, E)$ with $|A| = |B| = \Theta(\Delta)$, where the two bipartitions A and B are turned into two separate cliques. Observe that each such graph has a maximum independent set size of 2. We now claim that, for every non-edge $ab \in (V \times V) \setminus E(G)$ in the input graph G, there must exist an independent set I_i returned by the oracle such that $I_i = \{a, b\}$. This needs to be the case since otherwise the algorithm cannot distinguish between G and the graph $G' = G \cup \{ab\}$, i.e., G with the edge ab added, since all query responses are then consistent with both G and G'. However, since at most one non-edge can be learnt per query, and $\mathcal{G}$ contains many graphs with $\Omega(\Delta^2)$ non-edges, the result follows. Our actual lower bound is proved for randomized algorithms via an application of Yao's Lemma, which makes the argument slightly more complicated.

1.3 Recent Developments

Since a preprint of our work was released on arXiv in early 2024, Michael and Scott [23] improved some of our lower bounds by poly-logarithmic factors. They show that, for arbitrary adaptive randomized algorithms, $\Omega(\Delta^2 \log(n/\Delta)/\log\Delta)$ queries are needed, improving our $\Omega(\Delta^2)$ lower bound by a $\log(n/\Delta)/\log\Delta$ factor. They further strengthen this bound to $\Omega(\Delta^2 \log(n/\Delta))$ for non-adaptive randomized algorithms. For deterministic non-adaptive algorithms, they give a $\Omega(\Delta^3 \log n/\log\Delta)$ lower bound, improving over our $\Omega(\Delta^3/\log^2\Delta)$ lower bound.

Their improvements for randomized algorithms are obtained by using different and more involved graph constructions. Their lower bound for non-adaptive deterministic algorithms is obtained by establishing a connection between what we refer to as Δ-Query-Schemes (which are equivalent to non-adaptive deterministic algorithms) and cover-free families, which are well-studied mathematical objects. They give an improved lower bound for a cover-free family with certain parameters, which, by the novel connection identified, translates to a lower bound for deterministic non-adaptive algorithms.

1.4 Outline

We give our algorithms in Section 2 and our lower bounds in Section 3. We conclude with open problems in Section 4.

2 Algorithms

We give our randomized non-adaptive algorithm that executes $O(\Delta^2 \log n)$ queries in Subsection 2.1, and our deterministic non-adaptive algorithm that executes $O(\Delta^3 \log(\frac{n}{\Delta}))$ queries in Subsection 2.2. Both these algorithms require the maximum degree Δ as part of their inputs. In Subsection 2.3, we show how these algorithms can be turned into adaptive algorithms with a similar number of queries that do not require advanced knowledge of Δ.

2.1 Randomized Algorithm

Our algorithm, Algorithm 1, executes $\Theta(\Delta^2 \log n)$ queries on random subsets $V_i \subseteq V$, where each vertex is included in V_i with probability $\frac{1}{\Delta+1}$, and outputs every pair $uv \in V \times V$ as an edge of the input graph if $\{u, v\}$ is not contained in any maximal independent set I_i, for all i, returned by the oracle.

In the analysis, we prove that, for every non-edge $uv \in (V \times V) \setminus E(G)$ in the input graph G, both u and v are reported in any independent set I_i with probability $\Theta(\frac{1}{\Delta^2})$. Since $\Theta(\Delta^2 \log n)$ independent sets are computed, any non-edge will be detected with high probability, and by the union bound, this then applies to all non-edges.

Algorithm 1 Randomized graph reconstruction using a MIS oracle.

Require: Vertex set V, maximum degree Δ, large enough constant C
  for $i = 1 \dots C \cdot (\Delta+1)^2 \cdot \log n$ **do**
    $V_i \subseteq V$ random sample such that every $v \in V$ is included in V_i with probability $\frac{1}{\Delta+1}$
    $I_i \leftarrow \text{query}(V_i)$
  end for
  $E \leftarrow \{uv \in V \times V \ : \ \nexists i \text{ such that } \{u,v\} \subseteq I_i\}$
  return (V, E)

▶ **Theorem 1.** *Algorithm 1 is a non-adaptive randomized algorithm that executes $O(\Delta^2 \log n)$ MIS queries and correctly reconstructs a graph of maximum degree Δ with high probability.*

Proof. Let $G = (V, E)$ denote the input graph, Δ the maximum degree, and let $\overline{E} = (V \times V) \setminus E$ denote the set of non-edges. We will prove that, with high probability, every non-edge can be identified by the algorithm in that, for a non-edge $uv \in \overline{E}$, there exists an independent set I_i such that $\{u, v\} \in I_i$ with high probability.

Consider thus any non-edge $uv \in \overline{E}$. Then, for every i, the probability that both u and v are reported in independent set I_i is at least:

$$\Pr[u, v \in I_i] \geq \Pr[u, v \in V_i \text{ and } (\Gamma(u) \cup \Gamma(v)) \cap V_i = \varnothing] \ ,$$

since both u and v need to be included in a maximal independent set if they are isolated vertices in $G[V_i]$.

Next, observe that the events "$u, v \in V_i$" and "$(\Gamma(u) \cup \Gamma(v)) \cap V_i = \varnothing$" are independent since u and v are not adjacent and are thus not contained in $\Gamma(u) \cup \Gamma(v)$. Furthermore, observe that $|\Gamma(u) \cup \Gamma(v)| \leq 2\Delta$. Then:

$$\begin{aligned}
\Pr[u, v \in V_i \text{ and } (\Gamma(u) \cup \Gamma(v)) \cap V_i = \varnothing] &= \Pr[u, v \in V_i] \cdot \Pr[(\Gamma(u) \cup \Gamma(v)) \cap V_i = \varnothing] \\
&\geq \frac{1}{(\Delta+1)^2} \cdot (1 - \frac{1}{\Delta+1})^{2\Delta} \\
&\geq \frac{1}{(\Delta+1)^2} \cdot e^{-\frac{\frac{1}{\Delta+1}}{1-\frac{1}{\Delta+1}} \cdot 2\Delta} = \frac{1}{(\Delta+1)^2} \cdot \frac{1}{e^2} \ , \quad (1)
\end{aligned}$$

where we used the bound $1 - x \geq e^{-\frac{x}{1-x}}$, which holds for every $x < 1$.

Next, we compute the probability that both endpoints of the non-edge uv are never reported together:

$$\begin{aligned}
\Pr[\{u, v\} \not\subseteq I_i, \text{ for all } i] &= \prod_{i=1}^{C(\Delta+1)^2 \cdot \log n} \Pr[\{u, v\} \not\subseteq I_i] \\
&= \prod_{i=1}^{C(\Delta+1)^2 \cdot \log n} (1 - \Pr[u, v \in I_i]) \\
&\leq (1 - \frac{1}{(\Delta+1)^2} \cdot \frac{1}{e^2})^{C(\Delta+1)^2 \cdot \log n} \\
&\leq \exp\left(-\frac{1}{(\Delta+1)^2} \cdot \frac{1}{e^2} \cdot C(\Delta+1)^2 \cdot \log n\right) \leq \frac{1}{n^3} \ ,
\end{aligned}$$

where we used the inequality $1 + x \leq \exp(x)$, and the last inequality holds when C is chosen to be large enough.

By the union bound, the probability that the endpoints of at least one of the $O(n^2)$ non-edges are not reported in any independent set I_i is therefore at most $\frac{1}{n}$, which completes the proof. ◀

2.2 Deterministic Algorithm

Central to our deterministic algorithm is the notion of a Δ-*Query-Scheme* and a *Witness*:

▶ **Definition 2** (Witness). *Let V be a set of n vertices and let $2 \leq \Delta \leq n/2-1$ be an integer. Then, the tuple $(\{u,v\},\{w_1,\ldots,w_{2\Delta}\})$ with $u,v,w_1,\ldots,w_{2\Delta} \in V$ being distinct vertices is called a* Witness*, and we denote by $\mathcal{W}$ the set of all witnesses.*

As it will be important in the proof of Lemma 6, we give an upper bound on the number of witnesses:

▶ **Lemma 3.** *The number of witnesses $|\mathcal{W}|$ is bounded by:*

$$|\mathcal{W}| \leq n^2 \cdot (e \cdot \frac{n-2}{2\Delta})^{2\Delta} .$$

Proof. We use the bound $\binom{a}{b} \leq \left(e \cdot \frac{a}{b}\right)^b$ and obtain:

$$|\mathcal{W}| = \binom{n}{2} \cdot \binom{n-2}{2\Delta} \leq n^2 \cdot (e \cdot \frac{n-2}{2\Delta})^{2\Delta} .$$ ◀

▶ **Definition 4** (Δ-Query-Scheme). *Let V be a set of n vertices and let $2 \leq \Delta \leq n/2-1$ be an integer. The set $\mathcal{Q} = \{Q_1,\ldots,Q_\ell\}$ is denoted a* Δ-Query-Scheme *of size ℓ if, for every witness $(\{u,v\},\{w_1,\ldots,w_{2\Delta}\}) \in \mathcal{W}$, there exists a query $Q_i \in \mathcal{Q}$ such that:*

1. *$u,v \in Q_i$, and*
2. *$\{w_1,\ldots,w_{2\Delta}\} \cap Q_i = \varnothing$.*

In the following, we say that a query Q_i considers *a witness $W \in \mathcal{W}$ if Items 1 and 2 hold for Q_i and W.*

We show in Lemma 5 that a Δ-Query-Scheme of size ℓ immediately yields a non-adaptive deterministic query algorithm for GR for graphs of maximum degree Δ that executes ℓ queries. Our task is thus to design a Δ-Query-Scheme of small size, which we do in the proof of Lemma 6.

▶ **Lemma 5.** *Let $\mathcal{Q}$ be a Δ-Query-Scheme of size ℓ. Then, there exists a non-adaptive deterministic algorithm for GR that executes ℓ queries on graphs of maximum degree Δ.*

Proof. Let $G=(V,E)$ be the input graph, and let $\mathcal{Q}$ be a Δ-Query-Scheme of size ℓ. The algorithm executes every query $Q_i \in \mathcal{Q}$. Let I_i denote the query answer to Q_i.

We now claim that, for every non-edge $uv \in (V \times V) \setminus E$ in the input graph, there exists an independent set I_i such that $u,v \in I_i$. To see this, denote by $w_1,\ldots,w_{\deg(u)}$ the neighbours of u in G and by $w_{\Delta+1},\ldots,w_{\Delta+\deg(v)}$ the neighbours of v in G. Then, since $\mathcal{Q}$ is a Δ-Query-Scheme, there exists a query Q_i such that $u,v \in I_i$, but none of u's and v's neighbours are included. Hence, both u and v are necessarily included in I_i and the algorithm therefore observes a witness that proves that the edge uv does not exist in the input graph.

Since the argument applies to every non-edge, the algorithm learns all non-edges of the input graph and thus also learns all of the input graph's edges by complementing the set of non-edges. ◀

▶ **Lemma 6.** *There exists a Δ-Query-Scheme of size $O(\Delta^3 \log \frac{n}{\Delta})$.*

Proof. For an integer ℓ whose value we will determine later, let $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_\ell\}$ be such that, for every i, $Q_i \subseteq V$ is the subset of V obtained by including every vertex with probability $\frac{1}{\Delta+1}$. We use the probabilistic method and prove that $\mathcal{Q}$ is a Δ-Query-Scheme with positive probability, which in turn implies that such a scheme exists.

To this end, let $u, v, w_1 \ldots, w_{2\Delta} \in V$ be distinct vertices. Then, for any i, we obtain (the derivation is identical to Inequality 1 and therefore not repeated here)

$$\Pr[u, v \in Q_i \text{ and } w_1, \ldots, w_{2\Delta} \notin Q_i] = \frac{1}{(\Delta+1)^2} \cdot (1 - \frac{1}{\Delta+1})^{2\Delta} \geq \frac{1}{(\Delta+1)^2} \cdot \frac{1}{e^2} \ .$$

Furthermore, the probability that there does not exist a query Q_i, for any $i \in [\ell]$, such that $u, v \in Q_i$ and $w_1, \ldots, w_{2\Delta} \notin Q_i$ is at most:

$$\begin{aligned}\Pr[\nexists i \text{ such that } u, v \in Q_i \text{ and } w_1, \ldots, w_{2\Delta} \notin Q_i] &\leq \left(1 - \frac{1}{(\Delta+1)^2 e^2}\right)^\ell \\ &\leq \exp(-\frac{\ell}{(\Delta+1)^2 e^2}) \ .\end{aligned}$$

Hence, by the union bound over all witnesses $u, v, w_1, \ldots, w_{2\Delta}$ (see Lemma 3), the probability that there exists a witness that is not considered by the Δ-Query-Scheme is at most:

$$n^2 \cdot (e \cdot \frac{n-2}{2\Delta})^{2\Delta} \cdot \exp(-\frac{\ell}{(\Delta+1)^2 e^2}) \ .$$

For this probability to be strictly below 1, it is enough to set

$$\ell = \Theta(\Delta^3 \cdot \log(\frac{n}{\Delta})) \ ,$$

which in turn implies that such a scheme exists. ◀

Combining Lemma 5 and Lemma 6, we obtain the main result of this section.

▶ **Corollary 7.** *There exists a deterministic algorithm for GR that executes $O(\Delta^3 \log \frac{n}{\Delta})$ non-adaptive MIS queries for graphs with maximum degree Δ.*

2.3 Adaptive Algorithms without Knowledge of Δ

We will now show how the randomized and deterministic algorithms from the previous sections can be turned into adaptive algorithms with similar query complexity that do not require knowledge of Δ in order to operate. More specifically, let $\mathcal{A}$ be a non-adaptive query algorithm that, given an integer D, identifies with high probability in $R(D)$ rounds all non-edges uv such that $\deg(u) \leq D$ and $\deg(v) \leq D$ hold. It immediately follows from our analyses that both our randomized and deterministic query algorithms have this property when executed with parameter D instead of the true value of Δ. For our randomized algorithm, we have $R(D) = O(D^2 \log n)$, and, for our deterministic algorithm, we have $R(D) = O(D^3 \log(\frac{n}{D}))$. We will show how $\mathcal{A}$ can be turned into an adaptive algorithm that does not require knowledge of Δ and requires overall $O(\mathcal{R}(\Delta))$ rounds.

This is achieved via the doubling strategy displayed in Algorithm 2.

The algorithm uses the notation $\Delta(E')$, which is to be interpreted as the maximum degree in the graph spanned by the edges E'.

Algorithm 2 Adaptive algorithm that does not require advanced knowledge of Δ.

Require: Vertex set V, non-adaptive algorithm $\mathcal{A}$ that identifies all non-edges whose endpoints are of degree at most D, for some integer D
$D \leftarrow 1$, $E' \leftarrow V \times V$
while $\Delta(E') > D$ **do**
$D \leftarrow 2 \cdot D$
$F \leftarrow \mathcal{A}(D)$ {Set of non-edges identified}
$E' \leftarrow (V \times V) \setminus F$
end while
return (V, E')

▶ **Lemma 8.** *Let $\mathcal{A}$ be a non-adaptive MIS-query algorithm that, given an integer D, in $R(D)$ rounds identifies with high probability all non-edges $uv \in (V \times V) \setminus E$ of the input graph $G = (V, E)$ that have the property that $\deg(u) \leq D$ and $\deg(v) \leq D$. We assume that $R(D)$ is at least linear in D, i.e., $R(D) = \Omega(D)$. Then, there exists an adaptive algorithm for GR that succeeds with high probability, does not require advanced knowledge of Δ, runs in $O(R(\Delta))$ rounds, and requires $O(\log \Delta)$ rounds of adaptivity.*

Proof. Let $G = (V, E)$ denote the input graph and Δ the maximum degree.

We will argue that when exiting the last iteration of the while loop of the algorithm, the inequalities

$$\Delta \leq D < 2\Delta \tag{2}$$

hold. The lower bound $\Delta \leq D$ establishes correctness since the last run of $\mathcal{A}$ is executed with a guess $\Delta \leq D$, which implies that all non-edges are correctly identified. The upper bound is required in order to bound the query complexity of the algorithm.

We first argue the lower bound in Inequality 2. Observe that, in any iteration of the loop, $E' \supseteq E$ holds since F is a subset of the non-edges and $E' = (V \times V) \setminus F$. Thus, we have $\Delta(E') \geq \Delta(E) = \Delta$. When exiting the last iteration of the algorithm, we have $\Delta(E') \leq D$ since otherwise this would not be the last iteration of the algorithm. Combining these two inequalities yields $\Delta \leq D$.

Regarding the upper bound, we have just proved that, when exiting the last iteration, $\Delta \leq D$ holds. It is then also clear that a run with $\Delta \leq D < 2\Delta$ is executed since the guess D is doubled in each iteration. We claim that this run with $\Delta \leq D < 2\Delta$ is indeed the last iteration of the algorithm. Indeed, in this iteration, all non-edges F are correctly identified since $D \geq \Delta$, which implies that E' constitutes the edge set E of the input graph. This further implies that $\Delta(E') = \Delta$. The condition in the while loop then ensures that this is indeed the last iteration of the algorithm.

Last, regarding the runtime of the algorithm, let $D' = 2^i$ be the guess used in the last iteration. We then have that $\Delta \leq D' < 2\Delta$. Then, under the assumption that $R(D) = \Omega(D)$, we have:

$$R(2) + R(4) + \cdots + R(D') \leq 2 \cdot R(D') \ ,$$

which establishes the query complexity of the algorithm. Last, since the while loop is executed $O(\log \Delta)$ times, the algorithm requires only $O(\log \Delta)$ rounds of adaptivity. ◀

The previous lemma together with our randomized and deterministic algorithms from the previous sections yield the following corollary.

▶ **Corollary 9.** *There are randomized and deterministic adaptive MIS-query algorithms for GR that execute $O(\Delta^2 \cdot \log n)$ and $O(\Delta^3 \cdot \log \frac{n}{\Delta})$ queries, respectively, require $O(\log \Delta)$ rounds of adaptivity, and do not require advanced knowledge of Δ in order to operate. The randomized algorithm succeeds with high probability.*

3 Lower Bounds

In this section, we give three lower bound results. First, in Subsection 3.1, we consider the class of non-adaptive deterministic query algorithms and we prove that such algorithms require $\Omega(\Delta^3/\log^2 \Delta)$ queries. This result renders our deterministic query algorithm optimal, up to poly-logarithmic factors, and it also establishes a separation result between deterministic and randomized query algorithms, since, as demonstrated by our randomized query algorithm, $O(\Delta^2 \log n)$ non-adaptive randomized queries are sufficient.

Next, in Subsection 3.2, we show that $\Omega(\Delta^2)$ queries are needed for query algorithms that may be adaptive and randomized, and that $\Omega(\log n)$ queries are needed for such algorithms, even if the input graph is an n-vertex cycle.

3.1 Lower Bound for Non-adaptive Deterministic Algorithms

We will first show in Lemma 10 that any Δ-Query-Scheme must be of size at least $\Omega(\frac{\Delta^3}{\log^2 \Delta})$. Then, we argue in Lemma 11 that the queries executed by any non-adaptive deterministic query algorithm must constitute a $(\Delta - 1)$-Query-Scheme. These two lemmas together then imply our main result of this section as stated in Corollary 12, i.e., that non-adaptive deterministic query algorithms require $\Omega(\frac{\Delta^3}{\log^2 \Delta})$ queries.

▶ **Lemma 10.** *For every $\Delta \leq n^{2/3}$, every Δ-Query-Scheme is of size $\Omega(\frac{\Delta^3}{\log^2(\Delta)})$.*

Proof. Let C be a suitably large constant, and suppose that there exists a Δ-Query-Scheme $\mathcal{Q}$ of size $\ell = \frac{1}{6 \cdot C^2} \cdot \frac{\Delta^3}{\ln^2(\Delta)}$. We will show by contradiction that a Δ-Query-Scheme of this size does not exist. Furthermore, we define a relevant query size threshold t by $t := C \cdot \frac{n \ln(\Delta)}{\Delta}$.

Given $\mathcal{Q}$, we will first argue that there exists a pair of disjoint vertices $x, y \in V$ such that there is no query $Q \in \mathcal{Q}$ with $Q = \{x, y\}$, and there are at most $\Delta/2$ queries of size at most t that contain both x, y. Then, once we have established that such a pair $\{x, y\}$ exists then we use the probabilistic method to show that there exist 2Δ disjoint vertices $w_1, \ldots, w_{2\Delta}$ different to x, y such that the witness $(\{x, y\}, \{w_1, \ldots, w_{2\Delta}\}$ is not considered in $\mathcal{Q}$, a contradiction to the fact that $\mathcal{Q}$ is a Δ-Query-Scheme. This implies that a query scheme of size ℓ cannot exist.

Denote by $\mathcal{X}$ the set of subsets of V of size 2, i.e., $\mathcal{X} = \{\{u, v\} \ : \ u, v \in V, u \neq v\}$, where V denotes the set of n vertices of the input graph, and observe that $|\mathcal{X}| = \binom{n}{2}$.

First, observe that at most ℓ queries in $\mathcal{Q}$ are of size exactly 2. Hence, as long as $\ell \leq \binom{n}{2}/10$, at most a (1/10)-fraction of $\mathcal{X}$ is part of queries of size 2. Observe that, since the statement of the lemma assumes that $\Delta \leq n^{2/3}$, we have

$$\ell = \frac{1}{6 \cdot C^2} \cdot \frac{\Delta^3}{\ln^2(\Delta)} \leq \frac{\Delta^3}{6 \cdot C^2} \leq \frac{n^2}{6 \cdot C^2} \leq \frac{n^2}{40} \leq \frac{1}{10} \binom{n}{2} ,$$

where we assumed that C is large enough, and the last inequality holds for every $n \geq 2$. Observe that a query $Q = \{u, v\}$ of size 2 immediately considers all witnesses of the form $(\{u, v\}, \{w_1, \ldots, w_{2\Delta}\})$, for any vertices $(w_i)_{1 \leq i \leq 2\Delta}$. The previous argument shows that this can happen for at most a (1/10)-fraction of pairs $\{u, v\} \in \mathcal{X}$.

Next, we argue that at least a (1/4)-fraction of the pairs $\{u,v\} \in \mathcal{X}$ are such that $\{u,v\}$ is contained in at most $\frac{1}{2}\Delta$ queries of size at most t in $\mathcal{Q}$. To this end, observe that any query of size at most t contains at most $\binom{t}{2} \leq t^2$ distinct pairs $\{u,v\}$, and since there are overall ℓ queries, at most

$$\ell \cdot t^2 = \frac{1}{6C^2} \cdot \frac{\Delta^3}{\ln^2(\Delta)} \cdot \left(C \cdot \frac{n \ln(\Delta)}{\Delta} \right)^2 = \frac{1}{6} \cdot n^2 \cdot \Delta$$

pairs appear overall in all queries of size at most t. If less than a (1/4)-fraction of pairs $\{u,v\}$ were contained in at most $\frac{1}{2}\Delta$ queries of size at most t in $\mathcal{Q}$, then at least a (3/4)-fraction of pairs were contained in more than $\frac{1}{2}\Delta$ queries of size at most t. But this implies that the following inequality must hold:

$$(3/4) \cdot \binom{n}{2} \frac{1}{2}\Delta \leq \ell \cdot t^2 = \frac{1}{6} \cdot n^2 \cdot \Delta \ .$$

This inequality, however, does not hold for $n \geq 10$, a contradiction. Hence, we have proved that at least a (1/4)-fraction of pairs $\{u,v\} \in \mathcal{X}$ are such that $\{u,v\}$ is contained in at most $\frac{1}{2}\Delta$ queries of size at most t in $\mathcal{Q}$.

Consider thus a pair $\{u,v\} \in \mathcal{X}$ that is included in at most $\Delta/2$ queries of size at most t, and that is not included in a query of size 2. The arguments above ensure that such a pair exists. Denote by $Q_1, \ldots, Q_k$ the set of queries that contain both u and v, and suppose that $Q_1, \ldots, Q_j$ are the queries of size at most t (which implies $j \leq \Delta/2$). We now claim that there is a witness $W = (\{u,v\}, \{w_1, \ldots, w_{2\Delta}\})$, for some vertices $(w_i)_{1 \leq i \leq 2\Delta}$, that is not considered by the queries $Q_1, \ldots, Q_k$, contradicting the assumption that $\mathcal{Q}$ is a Δ-Query-Scheme, which then completes the proof.

The witness is constructed as follows. For $1 \leq i \leq j$, let $\tilde{w}_i \in Q_i \setminus \{u,v\}$ be any element. Recall that $\{u,v\}$ is not contained in any query of size exactly 2, hence, $\tilde{w}_i$ is well-defined. Furthermore, let $\tilde{W} = \{\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_j\}$. We observe that the elements $(\tilde{w}_i)_{1 \leq i \leq j}$ are not necessarily disjoint, and, hence, $|\tilde{W}| \in \{1, \ldots, j\}$. Next, let $R \subseteq V \setminus \left(\{u,v\} \cup \tilde{W}\right)$ be a random subset of size $2\Delta - |\tilde{W}|$. Then, our witness W is obtained as $W = \{\{u,v\}, \tilde{W} \cup R\}$, and we will prove in the following that, with positive probability, this witness is indeed not considered by $\mathcal{Q}$. This implies that there exists a witness that is not considered by $\mathcal{Q}$, which completes the proof.

Consider now any query Q_i, for $i \geq j+1$. Then, the probability that Q_i considers W is bounded as follows:

$$\begin{aligned}
&\Pr[Q_i \text{ considers } W] \leq \Pr[|Q_i \cap R| = \varnothing] \\
&\quad = \frac{n - |Q_i| - |\tilde{W}|}{n - 2 - |\tilde{W}|} \cdot \frac{n - |Q_i| - |\tilde{W}| - 1}{n - 2 - |\tilde{W}| - 1} \cdot \ldots \cdot \frac{n - |Q_i| - |\tilde{W}| - (2\Delta - |\tilde{W}|) + 1}{n - 2 - |\tilde{W}| - (2\Delta - |\tilde{W}|) + 1} \\
&\quad \leq \left(\frac{n - |Q_i| - |\tilde{W}|}{n - 2 - |\tilde{W}|} \right)^{2\Delta - |\tilde{W}|} \leq \left(\frac{n - |Q_i|}{n - 2} \right)^{2\Delta - |\tilde{W}|} \leq \left(\frac{n - |Q_i|}{n - 2} \right)^{1.5\Delta} \\
&\quad = \left(\frac{n - 2 + 2 - |Q_i|}{n - 2} \right)^{1.5\Delta} = \left(1 - \frac{|Q_i| - 2}{n - 2} \right)^{1.5\Delta} \leq \left(1 - \frac{|Q_i| - 2}{n} \right)^{1.5\Delta} \\
&\quad \leq \exp\left(-\frac{(|Q_i| - 2)1.5\Delta}{n} \right) \leq \exp\left(-1.5C \ln(\Delta) + \frac{3\Delta}{n} \right) \leq \frac{2}{\Delta^{1.5C}} \ .
\end{aligned}$$

Since C is a large enough constant, by the union bound, the probability that at least one query of the queries $Q_{j+1} \ldots, Q_k$ considers W is therefore strictly below 1 (recall that there are at most $k \leq \ell \leq \Delta^3$ queries, and $\Delta^3 \cdot \frac{2}{\Delta^{1.5C}} < 1$ holds, for large enough C). Hence, there exists a witness that is not considered by these queries, which completes the proof. ◀

▶ **Lemma 11.** *Let $\mathcal{A}$ be a non-adaptive deterministic query algorithm for GR on graphs of maximum degree Δ. Then, the queries executed by $\mathcal{A}$ form a $(\Delta-1)$-Query-Scheme.*

Proof. For the sake of a contradiction, suppose that there exists a sequence of queries $Q_1, \ldots, Q_k$ that does not form a $(\Delta-1)$-Query-Scheme but still allows the algorithm $\mathcal{A}$ to learn the input graph exactly. Since the queries do not form a $(\Delta-1)$-Query-Scheme, there exists a witness $W = (\{u, v\}, \{w_1, \ldots, w_{2\Delta-2}\})$ that is not considered by the queries. Consider now any two input graphs G_1 and G_2 of maximum degree Δ that have the following properties: u's neighbours are $w_1, \ldots, w_{\Delta-1}$, and v's neighbours are $w_\Delta, \ldots, w_{2\Delta-2}$. In G_1, there is also an edge between u and v, and in G_2 there is no edge between u and v. Observe that the degrees of u and v in G_1 and G_2 are Δ and $\Delta-1$, respectively. The maximum degrees in G_1 and G_2 are therefore at most Δ.

Now, we claim that, for every query Q_i, the oracle can respond with an independent set I_i that does not include both vertices u and v. Observe that the algorithm therefore cannot learn whether the edge uv exists since both G_1 and G_2 are consistent with all query answers. The algorithm therefore cannot distinguish between the two graphs G_1, G_2, which then completes the argument.

Since Q_i does not consider W, there exists a vertex $w \in \{w_1, \ldots, w_{2\Delta-2}\}$ such that $w \in Q_i$. Hence, the oracle can construct an independent set starting with vertex w (e.g., by running the Greedy maximal independent set algorithm where w is the first vertex picked), which implies that either u or v cannot be included in the independent set. This completes the proof. ◀

▶ Remark. The proof of the previous lemma assumes that the oracle can identify a witness not considered by the queries submitted by the algorithm. This is only possible if all queries are submitted simultaneously to the oracle. Observe that this is a valid assumption since we consider the class of non-adaptive algorithms, and such algorithms equally work when all queries are submitted simultaneously.

Combining Lemma 10 with Lemma 11, we obtain the main lower bound result of this section as a corollary:

▶ **Corollary 12.** *Every deterministic non-adaptive query algorithm for GR requires $\Omega(\Delta^3/\log^2(\Delta))$ queries.*

3.2 Lower Bounds for Adaptive Randomized Algorithms

We first prove that, even on an n-vertex cycle, any randomized adaptive algorithm requires $\Omega(\log n)$ queries to solve GR. Our proof is based on an indistinguishability argument: At least $\Omega(\log n)$ queries are needed so that, for each pair of vertices $u, v \in V$, different outcomes from the oracle are observed, which is needed to distinguish all possible cycles from each other.

▶ **Theorem 13.** *Every possibly randomized query algorithm with success probability strictly above $1/2$ for Graph Reconstruction using a Maximal Independent Set oracle on an n-vertex cycle requires $\Omega(\log n)$ queries.*

Proof. Denote by $\mathcal{C}$ the family of n-vertex cycles on the vertex set V with $|V| = n$, which serve as the input to our algorithm.

Let $\mathbf{A}$ be a randomized query algorithm that executes $\ell := \log_3(n) - 1$ rounds and that, on any input $C \in \mathcal{C}$, succeeds with probability strictly above $\frac{1}{2}$. Then, by Yao's lemma, there exists a deterministic query algorithm $\mathbf{A}_{\text{det}}$ that succeeds on strictly more than half of the inputs in $\mathcal{C}$ and also runs in ℓ rounds.

We first observe that, since $\mathbf{A}_{\text{det}}$ is deterministic, and we also assume that the oracle answers are deterministic, for every $C \in \mathcal{C}$, there exists a unique execution of $\mathbf{A}_{\text{det}}$, i.e., a sequence of query vertices $V_1, \ldots, V_\ell$, query answers $I_1, \ldots, I_\ell$, and output $C_{\text{out}} \in \mathcal{C}$ produced by the algorithm. We observe that C_{out} may be different from C if the algorithm errs on input C.

Denote by $\mathcal{C}_1 \subseteq \mathcal{C}$ the subset of inputs on which the algorithm succeeds, and let $\mathcal{C}_0 = \mathcal{C} \setminus \mathcal{C}_1$ denote the inputs on which the algorithm fails. We will now argue that, for every input $C \in \mathcal{C}_1$ on which the algorithm succeeds, there exists a unique input $C' \in \mathcal{C}_0$ on which the algorithm fails, i.e., there exists an injective mapping from $\mathcal{C}_1$ into $\mathcal{C}_0$, which implies that $|\mathcal{C}_1| \leq |\mathcal{C}_0|$. This is a contradiction to the fact that $\mathbf{A}_{\text{det}}$ is correct on strictly more than half of the instances in $\mathcal{C}$, which in turn implies that the algorithm $\mathbf{A}$ does not exist.

Let $C \in \mathcal{C}_1$ be an input on which $\mathbf{A}_{\text{det}}$ succeeds. Denote by $V_1, \ldots, V_\ell$ the vertices queried by the algorithm in the ℓ query rounds and by $I_1, \ldots, I_\ell$ the query responses. We associate the following complete ternary tree $\mathcal{T}$ with $\ell + 1$ layers to the query vertices and responses in an execution of $\mathbf{A}$:

- The root (layer 1) is labelled with $V = [n]$.
- For an internal node in layer $1 \leq i \leq \ell$ with label $U \subseteq V$, the node has three children with labels U_1, U_2, U_3 such that $U = U_1 \dot\cup U_2 \dot\cup U_3$, and

$$\begin{aligned} U_1 &= (U \cap V_i) \cap I_i \ , && \text{queried and reported} \\ U_2 &= (U \cap V_i) \setminus I_i \ , && \text{queried and not reported} \\ U_3 &= U \setminus V_i \ . && \text{not queried} \end{aligned}$$

Denote by $L_1, L_2, \ldots$ the leaves of $\mathcal{T}$ from left-to-right. We observe that, by construction, the leaves are disjoint and their union equals V, i.e., $\dot\cup_i L_i = V$. Consider now a leaf L_i that contains at least two vertices u, v. The vertices u, v are indistinguishable as they behaved exactly the same throughout the execution of the algorithm. Indistinguishable here means that the cycle C' obtained from the cycle C by swapping the positions of u and v leads to the exact same execution of the algorithm as the execution for C. The algorithm, however, fails on C' as the output produced is $C_{\text{out}} = C$. Hence, as long as there exists a leaf that contains at least two vertices, we can identify an input on which the algorithm fails and establish our injective mapping from $\mathcal{C}_1$ to $\mathcal{C}_0$. To avoid that there are no leaves that contain two vertices, the number of leaves in $\mathcal{T}$ must be at least n. Since $\mathcal{T}$ is ternary and of depth $\ell + 1$, we obtain that $\ell \geq \log_3(n)$ must hold, a contradiction to the assumption that $\ell = \log_3(n) - 1$. This completes the proof. ◀

Last, we give our $\Omega(\Delta^2)$ queries lower bound for graphs of maximum degree Δ. The key observation in our proof is that, for every non-edge uv in the input graph G, there must exist an oracle response maximal independent set that contains both vertices u and v, since, if the opposite was true then the algorithm could not distinguish between the input graph G and the graph $G \cup \{uv\}$.

▶ **Theorem 14.** *For any $\Delta > 0$, every possibly randomized query algorithm with success probability strictly greater than $1/2$ for Graph Reconstruction using a Maximal Independent Set oracle requires $\Omega(\Delta^2)$ queries on graphs of maximum degree Δ.*

Proof. For integers $N > 0$, let $\mathcal{H}_N$ denote the set of all bipartite graphs $H = (A, B, E)$ with $|A| = |B| = N$. Then, let $\mathcal{G}_N$ be the family of graphs obtained from $\mathcal{H}_N$ by turning the bipartitions A and B of each of its graphs $H = (A, B, E) \in \mathcal{H}_N$ into (disjoint) cliques.

Let $\mathbf{A}$ denote a randomized query algorithm that succeeds with probability strictly above $1/2$ on each input $G \in \mathcal{G}_N$, and for the sake of a contradiction, we assume that $\mathcal{A}$ executes at most $\ell = N^2 - 1$ queries. Then, by Yao's lemma, there exists a deterministic query algorithm $\mathcal{A}_{\text{det}}$ that also executes at most ℓ queries and succeeds on strictly more than half of the inputs in $\mathcal{G}_N$.

First, observe that $\alpha(G) \leq 2$, for every $G \in \mathcal{G}_N$ since at most one vertex from A and at most one vertex from B can be included in any independent set. Hence, every independent set reported by $\mathbf{A}_{\text{det}}$ on any of the input graphs $G \in \mathcal{G}_N$ is of size at most 2.

Denote by $\mathcal{G}_N^1 \subseteq \mathcal{G}_N$ the subset of inputs on which $\mathcal{A}_{\text{det}}$ succeeds, and let $\mathcal{G}_N^0 = \mathcal{G}_N \setminus \mathcal{G}_N^1$. We will show that there exists an injective map from $\mathcal{G}_N^1$ to $\mathcal{G}_N^0$, which implies that $|\mathcal{G}_N^1| \leq |\mathcal{G}_N^0|$, a contradiction to the fact that $\mathcal{A}_{\text{det}}$ succeeds on strictly more than half of the instances. This in turn implies that algorithm $\mathbf{A}$ does not exists, which establishes the theorem.

To this end, let $G \in \mathcal{G}_N^1$ be any instance and consider the execution of $\mathcal{A}_{\text{det}}$ on G, i.e., let $V_1, \ldots, V_\ell$ denote the queries submitted in the ℓ query rounds, let $I_1, \ldots, I_\ell$ denote the query responses, and let $G_{\text{out}} = G$ denote the output produced by the algorithm.

Let

$$F = \{(a,b) \in A \times B \mid \{a,b\} \neq I_i, \text{ for all } i\} ,$$

i.e., the set of pairs of vertices a, b that do not constitute a response from the oracle. We observe that every graph G' obtained from G by flipping the edge ab, i.e., by introducing ab in G' in case it is not contained in G or by removing it from G' in case it is contained in G, leads to the exact same execution of $\mathbf{A}_{\text{det}}$ as all oracle answers are consistent with both G and G'. The algorithm however fails on G' since the answer produced by the execution is $G_{\text{out}} = G$. Hence, as long as $F \neq \varnothing$, we can identify a graph G' on which the algorithm fails and map the input G to G' in our injective map. To ensure that $F = \varnothing$, we require at least $|A| \cdot |B| = N^2$ queries. This however is a contradiction to the assumption that only $\ell = N^2 - 1$ queries are executed.

Last, the result follows by observing that the maximum degree Δ of any graph in $\mathcal{G}_N$ is $2N - 1$, which implies that at least $(\frac{\Delta+1}{2})^2$ query rounds are required. ◀

We remark that Theorem 14 also shows that $\Omega(m)$ queries are needed for reconstructing graphs on m edges. Observe that every graph of the family of graphs used in Theorem 14 has $O(\Delta^2)$ edges.

4 Conclusion

In this paper, we initiated the study of the GR problem using an MIS oracle. We gave a non-adaptive randomized algorithm that reconstructs a graph with maximum degree Δ using $O(\Delta^2 \log n)$ queries, and a non-adaptive deterministic query algorithm that uses $O(\Delta^3 \log(\frac{n}{\Delta}))$ queries. Both these algorithms require advanced knowledge of Δ in order to operate, which is unavoidable. We showed that both algorithms can be turned into adaptive algorithms with $O(\log \Delta)$ rounds of adaptivity and a similar number of queries that do not require advanced knowledge of Δ. We also proved that, for adaptive randomized algorithms, $\Omega(\Delta^2)$ queries are necessary, and that such algorithms require $\Omega(\log n)$ queries even if the input graph is an n-vertex cycle. Furthermore, we showed that non-adaptive deterministic query algorithms require $\Omega(\Delta^3/\log^2(\Delta))$ queries, which renders our deterministic algorithm optimal up to poly-log factors.

While, up to lower order terms, the MIS query complexity of GR is settled for randomized algorithms and for non-adaptive deterministic algorithms, it is unclear whether there are adaptive deterministic algorithms that are stronger than non-adaptive deterministic algorithms. More concretely, is there an adaptive deterministic query algorithm that requires fewer than $O(\Delta^3 \log(\frac{n}{\Delta}))$ queries?

References

1 Hasan Abasi and Nader H. Bshouty. On learning graphs with edge-detecting queries. In Aurélien Garivier and Satyen Kale, editors, *Algorithmic Learning Theory, ALT 2019, 22-24 March 2019, Chicago, Illinois, USA*, volume 98 of *Proceedings of Machine Learning Research*, pages 3–30. PMLR, 2019. URL: http://proceedings.mlr.press/v98/abasi19a.html.

2 Mikkel Abrahamsen, Greg Bodwin, Eva Rotenberg, and Morten Stöckel. Graph reconstruction with a betweenness oracle. In Nicolas Ollinger and Heribert Vollmer, editors, *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, volume 47 of *LIPIcs*, pages 5:1–5:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.STACS.2016.5.

3 Raghavendra Addanki, Andrew McGregor, and Cameron Musco. Non-adaptive edge counting and sampling via bipartite independent set queries. In Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman, editors, *30th Annual European Symposium on Algorithms, ESA 2022, September 5-9, 2022, Berlin/Potsdam, Germany*, volume 244 of *LIPIcs*, pages 2:1–2:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ESA.2022.2.

4 Kook Jin Ahn, Graham Cormode, Sudipto Guha, Andrew McGregor, and Anthony Wirth. Correlation clustering in data streams. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2237–2246. JMLR.org, 2015. URL: http://proceedings.mlr.press/v37/ahn15.html.

5 Noga Alon and Vera Asodi. Learning a hidden subgraph. *SIAM J. Discret. Math.*, 18(4):697–712, 2005. doi:10.1137/S0895480103431071.

6 Noga Alon, Richard Beigel, Simon Kasif, Steven Rudich, and Benny Sudakov. Learning a hidden matching. *SIAM J. Comput.*, 33(2):487–501, 2004. doi:10.1137/S0097539702420139.

7 Dana Angluin and Jiang Chen. Learning a hidden graph using o(logn) queries per edge. *J. Comput. Syst. Sci.*, 74(4):546–556, 2008. doi:10.1016/J.JCSS.2007.06.006.

8 Sepehr Assadi, Christian Konrad, Kheeran K. Naidu, and Janani Sundaresan. O(log log n) passes is optimal for semi-streaming maximal independent set. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 847–858. ACM, 2024. doi:10.1145/3618260.3649763.

9 Sepehr Assadi and Shay Solomon. When algorithms for maximal independent set and maximal matching run in sublinear time. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 17:1–17:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICS.ICALP.2019.17.

10 Paul Beame, Sariel Har-Peled, Sivaramakrishnan Natarajan Ramamoorthy, Cyrus Rashtchian, and Makrand Sinha. Edge estimation with independent set oracles. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 38:1–38:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICS.ITCS.2018.38.

11 Richard Beigel, Noga Alon, Simon Kasif, Mehmet Serkan Apaydin, and Lance Fortnow. An optimal procedure for gap closing in whole genome shotgun sequencing. In Thomas Lengauer, editor, *Proceedings of the Fifth Annual International Conference on Computational Biology, RECOMB 2001, Montréal, Québec, Canada, April 22-25, 2001*, pages 22–30. ACM, 2001. doi:10.1145/369133.369152.

12 Lidiya Khalidah binti Khalil and Christian Konrad. Constructing large matchings via query access to a maximal matching oracle. In Nitin Saxena and Sunil Simon, editors, *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2020, December 14-18, 2020, BITS Pilani, K K Birla Goa Campus, Goa, India (Virtual Conference)*, volume 182 of *LIPIcs*, pages 26:1–26:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.FSTTCS.2020.26.

13 Uriel Feige. On sums of independent random variables with unbounded variance and estimating the average degree in a graph. *SIAM Journal on Computing*, 35(4):964–984, 2006. doi:10.1137/S0097539704447304.

14 David Galvin. Three tutorial lectures on entropy and counting, 2014. arXiv:1406.7872.

15 Mohsen Ghaffari. An improved distributed algorithm for maximal independent set. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 270–277. SIAM, 2016. doi:10.1137/1.9781611974331.CH20.

16 Mohsen Ghaffari, Themis Gouleakis, Christian Konrad, Slobodan Mitrovic, and Ronitt Rubinfeld. Improved massively parallel computation algorithms for mis, matching, and vertex cover. In Calvin Newport and Idit Keidar, editors, *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018, Egham, United Kingdom, July 23-27, 2018*, pages 129–138. ACM, 2018. doi:10.1145/3212734.3212743.

17 Oded Goldreich and Dana Ron. Approximating average parameters of graphs. In Josep Díaz, Klaus Jansen, José D. P. Rolim, and Uri Zwick, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 363–374, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. doi:10.1007/11830924_34.

18 Vladimir Grebinski and Gregory Kucherov. Optimal query bounds for reconstructing a hamiltonian cycle in complete graphs. In *Fifth Israel Symposium on Theory of Computing and Systems, ISTCS 1997, Ramat-Gan, Israel, June 17-19, 1997, Proceedings*, pages 166–173. IEEE Computer Society, 1997. doi:10.1109/ISTCS.1997.595169.

19 Sampath Kannan, Claire Mathieu, and Hang Zhou. Graph reconstruction and verification. *ACM Trans. Algorithms*, 14(4):40:1–40:30, 2018. doi:10.1145/3199606.

20 Christian Konrad, Kheeran K. Naidu, and Arun Steward. Maximum matching via maximal matching queries. In Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté, editors, *40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023, March 7-9, 2023, Hamburg, Germany*, volume 254 of *LIPIcs*, pages 41:1–41:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.STACS.2023.41.

21 Claire Mathieu and Hang Zhou. A simple algorithm for graph reconstruction. In Petra Mutzel, Rasmus Pagh, and Grzegorz Herman, editors, *29th Annual European Symposium on Algorithms, ESA 2021, September 6-8, 2021, Lisbon, Portugal (Virtual Conference)*, volume 204 of *LIPIcs*, pages 68:1–68:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.ESA.2021.68.

22 Andrew McGregor and Rik Sengupta. Graph reconstruction from random subgraphs. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 96:1–96:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ICALP.2022.96.

23 Lukas Michel and Alex Scott. Lower bounds for graph reconstruction with maximal independent set queries, 2024. doi:10.48550/arXiv.2404.03472.

24 Lev Reyzin and Nikhil Srivastava. Learning and verifying graphs using queries with a focus on edge counting. In Marcus Hutter, Rocco A. Servedio, and Eiji Takimoto, editors, *Algorithmic Learning Theory*, pages 285–297, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. doi:10.1007/978-3-540-75225-7_24.

25 Guozhen Rong, Wenjun Li, Yongjie Yang, and Jianxin Wang. Reconstruction and verification of chordal graphs with a distance oracle. *Theoretical Computer Science*, 859:48–56, 2021. doi:10.1016/j.tcs.2021.01.006.

26 Guozhen Rong, Yongjie Yang, Wenjun Li, and Jianxin Wang. A divide-and-conquer approach for reconstruction of C?5-free graphs via betweenness queries. *Theoretical Computer Science*, 917:1–11, 2022. doi:10.1016/j.tcs.2022.03.008.

27 Václav Rozhon and Mohsen Ghaffari. Polylogarithmic-time deterministic network decomposition and distributed derandomization. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 350–363. ACM, 2020. doi:10.1145/3357713.3384298.

A Independent Set Queries for Graph Reconstruction

For completeness, we will now prove that the number of Independent Set queries (or, in fact, any type of query that yields a binary answer) needed for GR on graphs of maximum degree $\Delta \in \Omega(\log n)$ is $\Omega(n\Delta \log(\frac{n}{\Delta}))$.

Angluin and Chen [7] observe that, since an independent set query has a binary output, at least $\log|\mathcal{G}|$ queries are needed to distinguish any two graphs in a graph family $\mathcal{G}$.

We will now prove that the number of n-vertex graphs with maximum degree Δ is $2^{\Omega(n\Delta\log(n/\Delta))}$, which, by the previous argument, implies that $\Omega(n\Delta\log(n/\Delta))$ Independent Set queries are needed to distinguish these graphs.

Our proof uses entropy-based arguments. We refer the reader to [14] for an excellent overview of how entropy is connected to counting problems.

▶ **Lemma 15.** *The number of bipartite $2n$-vertex graphs with bipartitions A and B each of size n and with maximum degree $\Delta = \Omega(\log n)$ is at least:*

$$2^{\frac{1}{2}n\Delta\log(\frac{n}{\Delta})-2} .$$

Proof. We consider the following probabilistic process: Let $G = (A, B, E)$ be a bipartite $2n$-vertex graph with $|A| = |B| = n$ obtained by inserting every potential edge $ab \in A \times B$ into G with probability $\frac{\Delta}{2n}$. Denote by E the indicator random variable of the event that G does not contain a vertex of degree larger than Δ.

We will now bound the quantity $|\text{range}(G|E = 1)|$ from below, which constitutes a set of bipartite graphs with maximum degree Δ.

To this end, first, observe that:

$$\log(|\text{range}(X|E = 1)|) \geq H(X|E = 1) ,$$

which implies that it is enough to bound $H(X|E = 1)$. To bound this quantity, we apply the chain rule for entropy twice on the expression $H(XE)$:

$$H(XE) = H(X) + H(E|X) , \text{ and}$$
$$H(XE) = H(E) + H(X|E) = H(E) + \Pr[E = 0]H(X|E = 0) + \Pr[E = 1]H(X|E = 1) ,$$

which implies:

$$\begin{aligned} H(X|E=1) &= \frac{H(X)+H(E|X)-H(E)-\Pr[E=0]H(X|E=0)}{\Pr[E=1]} \\ &\geq H(X)+H(E|X)-H(E)-\Pr[E=0]H(X|E=0) \\ &\geq H(X)-1-\Pr[E=0]H(X|E=0)\ , \end{aligned} \tag{3}$$

using the fact that entropy is non-negative, and that the inequality $H(E|X) \leq H(E) \leq 1$ holds.

Before bounding Inequality 3 further, we first prove that $\Pr[E=0]$ is small and we give a bound on $H(X)$.

To see that $\Pr[E=0]$ is small, consider any vertex $v \in A \cup B$. Then, the expected degree of v in G is $\Delta/2$, and, by a Chernoff bound, the probability that the degree of v is larger than Δ is at most $\frac{1}{2 \cdot n^4}$ (using the assumption that $\Delta = \Omega(\log n)$). By the union bound, the probability that there exists a vertex of degree larger than Δ is thus at most $\frac{1}{n^3}$, or, equivalently, $\Pr[E=0] \leq \frac{1}{n^3}$.

Next, we bound $H(X)$. Since each of the n^2 potential edges is included in G independently of all other edges, we obtain:

$$\begin{aligned} H(X) &= n^2 \cdot H_2(\frac{\Delta}{2n}) \\ &\geq n^2 \log(\frac{2n}{\Delta})\frac{\Delta}{2n} \geq \frac{1}{2} n\Delta \log(\frac{n}{\Delta})\ , \end{aligned}$$

where we bounded the binary entropy function by considering only one of its the two terms.

We are now ready to further simplify Inequality 3, which then yields the result:

$$\begin{aligned} H(X|E=1) &\geq H(X)-1-\Pr[E=0]H(X|E=0) \\ &\geq \frac{1}{2} n\Delta \log(\frac{n}{\Delta}) - 1 - \frac{1}{n^3} \cdot \log(\text{range}(X|E=0)) \\ &\geq \frac{1}{2} n\Delta \log(\frac{n}{\Delta}) - 1 - \frac{1}{n^3} \cdot n^2 \\ &\geq \frac{1}{2} n\Delta \log(\frac{n}{\Delta}) - 1 - \frac{1}{n} \\ &\geq \frac{1}{2} n\Delta \log(\frac{n}{\Delta}) - 2\ . \end{aligned}$$

◀

▶ **Corollary 16.** *The number of Independent Set queries needed for GR on n-vertex graphs of maximum degree $\Delta = \Omega(\log n)$ is $\Omega(n\Delta \log(\frac{n}{\Delta}))$.*

Error-Correcting Graph Codes

Swastik Kopparty
Department of Mathematics and Department of Computer Science, University of Toronto, Canada

Aditya Potukuchi
Department of Electrical Engineering and Computer Science, York University, Toronto, Canada

Harry Sha[1]
Department of Mathematics and Department of Computer Science, University of Toronto, Canada

Abstract

In this paper, we construct *Error-Correcting Graph Codes*. An error-correcting graph code of distance δ is a family C of graphs, on a common vertex set of size n, such that if we start with any graph in C, we would have to modify the neighborhoods of at least δn vertices in order to obtain some other graph in C. This is a natural graph generalization of the standard Hamming distance error-correcting codes for binary strings.

Yohananov and Yaakobi were the first to construct codes in this metric. We extend their work by showing

1. Combinatorial results determining the optimal rate vs distance trade-off nonconstructively.
2. Graph code analogues of Reed-Solomon codes and code concatenation, leading to positive distance codes for all rates and positive rate codes for all distances.
3. Graph code analogues of dual-BCH codes, yielding large codes with distance $\delta = 1 - o(1)$. This gives an explicit "graph code of Ramsey graphs".

Several recent works, starting with the paper of Alon, Gujgiczer, Körner, Milojević, and Simonyi, have studied more general graph codes; where the symmetric difference between any two graphs in the code is required to have some desired property. Error-correcting graph codes are a particularly interesting instantiation of this concept.

2012 ACM Subject Classification Mathematics of computing → Coding theory

Keywords and phrases Graph codes, explicit construction, concatenation codes, tensor codes

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.67

Funding *Swastik Kopparty*: Research supported by an NSERC Discovery Grant.
Aditya Potukuchi: Research supported by an NSERC Discovery Grant.

Acknowledgements We are grateful to Mike Saks, Shubhangi Saraf and Pat Devlin for valuable discussions.

1 Introduction

In this paper, we studi *Error-Correcting Graph Codes*. These are large families of undirected graphs on the same vertex set such that any two graphs in the family are "far apart" in a natural graph distance. Informally, the graph distance between two graphs on the same vertex set of size n measures the minimum number of vertices one needs to delete to make the resulting graphs identical (not just isomorphic). This can also be thought of as (1) the number of vertices whose neighborhoods one has to modify to go from one graph to another, (2) the vertex cover number of the symmetric difference of the two graphs, or (3) n minus the largest independent set in the symmetric difference of the two graphs.

[1] Optional footnote, e.g. to mark corresponding author

© Swastik Kopparty, Aditya Potukuchi, and Harry Sha;
licensed under Creative Commons License CC-BY 4.0

Definition (Graph Distance). Given two graphs G and H on vertex set $[n]$, the graph distance $\mathrm{d}_{\mathrm{graph}}(G, H)$ is the size of the smallest set $S \subseteq [n]$ such that $G[\overline{S}] = H[\overline{S}]$.

This is a very natural metric and encompasses deep information about graphs. For example, note the following two simple facts (1) the graph distance of a graph from the empty graph is n minus the independence number of the graph. (2) the graph distance of a graph from the complete graph is n minus the clique number of the graph. Thus the answer to the question: "how far can a graph be from both the empty graph and the complete graph?" is precisely the question of finding the right bound for the diagonal Ramsey numbers; the answer is $n - O(\log n)$.

Additionally, the notion of graph distance arises in the definition of node differential privacy (see for example [5, 8, 11, 9]). One instantiation of this setting is a graph encoding a social network where vertices correspond to people and edges correspond to social connections. The goal for node differential privacy is to design an algorithm $\mathcal{A}$ that approximately computes certain statistics of the graph (such as counting edges, triangles, and connected components) while maintaining each individual's privacy. Here, privacy is ensured by requiring that the output distribution on a certain graph does not change by much with any one vertex is deleted. In other words, for any graphs G, H of graph distance 1, the output distribution of $\mathcal{A}$ on the G and H should be similar. Graph distances of greater than 1 are then considered in the continual release model where the graph varies over time as studied in [9].

Codes in the graph-distance metric were initially studied by Yohananov, Efron, and Yaakobi [19, 18], who gave several optimal and near-optimal constructions in a variety of parameters. Their setting allows for arbitrary symbols $\alpha \in_q$ to be written on the edges, and the graph is allowed to have self-loops. We provide several new constructions for the binary setting, and the graph is not allowed to have self-loops. We show that random codes are asymptotically optimal codes in this metric and so focus our attention on explicit constructions.

Error-correcting graph codes also fall into the general framework of graph codes defined by Alon, Gujgiczer, Körner, Milojević, and Simonyi [3], where for a fixed family $\mathcal{F}$ of graphs, one seeks a large code C of graphs on the same n-vertex set such that the symmetric difference of any two graphs in C does not lie in $\mathcal{F}$. This class of problems was studied for a wide variety of natural $\mathcal{F}$ in a number of recent works [3, 2, 1]. As discussed in [2], for a suitable choice of $\mathcal{F}$, graph codes become equivalent to classical Hamming distance codes.

Finally, we note that error-correcting codes are pseudorandom objects, and the connection to Ramsey graphs suggests that error-correcting graph codes might be closely related to pseudorandom graphs. Thus, the problem of studying and explicitly constructing a pseudorandom family of pseudorandom graphs is interesting in its own right.

1.1 Related work

Similar to the Hamming setting, we briefly define the dimension, rate, and distance of a graph code C on n vertices where each edge is allowed to have an arbitrary symbol $\alpha \in \mathbb{F}_q$ written on it.

- **Dimension.** $k = \log_q(|C|)$.
- **Rate.** $R = \log_q(|C|)/\binom{n}{2}$.
- **Distance.** The distance of a code is the largest d such that $\mathrm{d}_{\mathrm{graph}}(G, H) \geq d$ for each $G, H \in C$ such that $G \neq H$. The relative distance δ is d/n.

Unless specified otherwise , we will always be interested in asymptotics of the above parameters as $n \to \infty$.

As noted in [19], an argument similar to the Singleton bound states that for a graph code of dimension k, and distance d, we have

$$k \leq \binom{n-d+1}{2}. \tag{1}$$

In terms of and rate R, and relative distance δ, we have

$$R \leq (1-\delta)^2 + o(1). \tag{2}$$

A code is called optimal if it $R \sim (1-\delta)^2$, equivalently, $\delta \sim 1 - \sqrt{R}$.

The relevant constructions in [19], [18] are summarized in Table 1. Note that all of the constructions in Table 1 are for undirected graphs where self-loops are allowed. One can impose n linear constraints to get codes with no self-loops (zeros on the diagonals of the adjacency matrix). Since the block length of these codes is $\binom{n}{2}$, adding these these constraints changes the rate by $o(1)$.

Table 1 Summary of constructions in [19], and [18].

Name	δ	Field	Tradeoff
[19] (Construction 1)	$= 2/n$	$_2$	Optimal
[19] (Construction 3)	≤ 1	$_q, q \geq n-1$	Optimal
[19] (Construction 5)	$\leq 1/2$	$_2$	$R = 1 - 2\delta$
[18] (Construction 1)	$= 3/n$	$_2$	Optimal
[18] (Construction 2)	$= 4/n$	$_2$	Optimal

Construction 5 of [19] leverages a connection to symmetric array codes and construction by Schmidt [15]. The trade-off achieved, $R \sim 1 - 2\delta$ is close to optimal for δ very small.

Construction 3 of [19] translates Hamming distance into graph distance using the tensor product. Although this construction doesn't directly imply anything about binary codes due to the requirement on field size, a similar code will be an important ingredient in our constructions.

1.2 Results

Our main results are:

1. We show that there are optimal binary codes for any constant $\delta \in (0,1)$.
2. We give constructions of graph codes that have positive constant R for all constant $\delta < 1$. In particular, we give a quasi-polynomial time explicit construction achieving $\delta = 1 - O(R^{1/4})$, while optimal codes have $\delta = 1 - R^{1/2}$. We also give an explicit construction with $\delta = 1 - O(R^{1/6})$, and a strongly explicit construction with $\delta = 1 - O(R^{1/8})$. Although these codes are not optimal, they are the first binary error-correcting graph codes achieving $\delta > 1/2$ with a constant rate.
3. We give (strongly) explicit constructions of graph codes with very high $\delta = 1 - O(n^{-\epsilon})$ and $R = \Omega(n^{\epsilon - 1/2})$ for constant $\epsilon > 0$. This gives a "graph code of Ramsey graphs" as will be discussed later.

Independent work

Pat Devlin and Cole Franks [6] independently proposed the study of graph error-correcting codes under this metric, determined the optimal R vs δ tradeoff, and gave some weaker explicit constructions of graph codes that worked for certain ranges of R and δ.

1.3 Techniques

We now discuss our techniques. We will often specify graphs by their adjacency matrices, viewed as matrices with $_2$ entries.

Our nonconstructive existence result is a straightforward application of the probabilistic method. We consider a uniformly random $_2$-linear subspace of the $_2$-linear space of symmetric 0-diagonal $n \times n$ matrices (i.e., the space of all adjacency matrices of graphs); this turns out to give a good graph code with optimal R vs δ tradeoff. Such graph codes can be specified by an $_2$ basis for it. We say that a construction is *explicit* if this basis can be produced in poly(n) time. We say it is *strongly explicit* if, given (i, j, k), the (i, j) entry of the k'th basis element can be computed in poly$(\log(n))$ time.

To get asymptotically good codes for any constant $\delta \in (0, 1)$ we take a longer detour.

1. We start with a slight variation of [19] (Construction 3), which gives a way to get a good graph code from a classical Hamming-distance linear code $C \subseteq_2^n$. We first consider the tensor code $C \otimes C$, where the elements are matrices all of whose rows and columns are codewords of C. A-priori, C could contain matrices that are neither symmetric, not have a 0 diagonal. But interestingly, if we consider the set C^* of all matrices in $C \otimes C$ that are symmetric and have 0 diagonal, then C^* is a linear space with quite large dimension. In particular, if the classical Hamming distance code C has positive rate, then so does the graph code C^*. We call this construction $C^* = \text{STCZD}(C)$ (Symmetric Tensor Code with Zero Diagonals).
   It turns out that if C has good relative distance (in the Hamming metric), then STCZD(C) has good distance in the graph metric. However the relative graph distance of STCZD(C) such a code is bounded by the relative distance of C – and since C is a binary code, this is at most $1/2$.
2. Now, we bring in another idea from the Hamming code world: code concatenation. Instead of constructing a graph code of symmetric zero-diagonal matrices over $_2$, we instead construct a "large-alphabet graph code" of symmetric zero-diagonal matrices over $_q$ for some large $q = 2^t$ and then try to reduce the alphabet size down to 2 by replacing the q-ary symbols with $_2$-matrices with suitable properties.
   Applying the analog of STCZD to a large alphabet code allows one to get large-alphabet graph codes with large δ, approaching 1 (since over large alphabets Hamming distance codes can have length approaching 1). Using Reed-Solomon codes as these large alphabet codes also allows us to make the STCZD construction strongly explicit. Furthermore, when applied to Reed-Solomon codes, these codes have a natural direct description: these are the evaluation tables of low-degree bivariate polynomials $P(X, Y)$ on product sets $S \times S$ that are (1) symmetric (to get a symmetric matrix), and (2) multiples of $(X - Y)^2$ (to get zero diagonal).
3. What remains now is to develop the right kind of concatenation so that the resulting graph code has good distance. This turns out to be subtle and requires an "inner code" with a stronger "directed graph distance" property with δ nearly 1. Fortunately, this inner code we seek is of very slowly growing size, and we may find this by brute force search. This concludes our description of our explicit construction of graph codes with δ approaching 1 and positive constant R.

Finally, we discuss our constructions for very high distances, $\delta = 1 - o(1)$. In this regime, as mentioned earlier, this is related to constructions of Ramsey graphs, a difficult problem in pseudorandomness with a long history. Our constructions work up to $\delta = 1 - \Omega\left(\frac{1}{\sqrt{n}}\right)$; concretely, we get a large linear space of graphs such that all graphs in the family have no

clique or independent set of size $\Omega(\sqrt{n})$. The construction is based on polynomials over finite fields of characteristic 2: When $n = 2^t$, we consider a linear space of certain low degree univariate polynomials $f(X)$ over $_{2^t}$, and create the $_2$ matrix with rows and columns indexed by $_{2^t}$ whose x, y entry is $\mathsf{Tr}(f(x+y))$. Here Tr is the finite field trace map from $_{2^t}$ to $_2$. The use of Tr of polynomials is inspired by the construction of dual-BCH codes. We then show that any such matrix has no large clique or independent set unless $\mathsf{Tr} \circ f$ is identically 0 or identically 1 (corresponding to the empty and complete graphs respectively). The proof uses the Weil bounds on character sums and a Fourier analytic approach to bound the independence number for the graphs. Our constructions are listed in Table 2.

Table 2 A list of constructions of error-correcting graph codes in this paper. All except Concatenated RS Tensor Codes are explicit. Note that the last four constructions are interesting in the regime where δ is close to 1.

Name	Approximate Tradeoffs	Strongly Explicit?
Random Linear Codes (Proposition 4)	$R = (1-\delta)^2 - o(1)$	No
Concatenated RS Tensor Codes (Code 19)	$R = (1-\sqrt{\delta})^4 - o(1)$	No
Double Concatenated RS Tensor Codes	$R = (1-\delta^{1/3})^6 - o(1)$	No
Triple Concatenated RS Tensor Codes (Code 22)	$R = (1-\delta^{1/4})^8 - o(1)$	Yes
Dual BCH Codes (Code 28)	$R = \log(n)(1-\delta)/\sqrt{n}$	Yes

1.4 Concluding thoughts and questions

The most interesting question in this context is to obtain explicit constructions of graph codes with optimal R vs δ tradeoff. While we have several constructions achieving nontrivial parameters in various regimes, it would even be interesting to get the right asymptotic behavior for the endpoints with δ approaching 1. The setting of large δ (including $\delta = 1 - o(1)$) seems especially challenging, given the connection with the notorious problem of constructing Ramsey graphs.

Another interesting question is to get decoding algorithms for graph codes. For a certain graph code C, if we are given a graph that is promised to be close in graph distance to some graph G in C. Then, can we efficiently find G?

A more general context relevant to error-correcting graph codes is the error correction of strings under more general error patterns. Suppose we have a collection of subsets $S_i \subseteq [m]$ for $i \in [t]$, where $\bigcup_i S_i = [m]$. These S_i denote the corruption zones; a single "corruption" of a string $z \in \{0,1\}^m$ entails, for some $i \in [t]$, changing $z|_{S_i}$ to something arbitrary in $\{0,1\}^{S_i}$. We want to design a code $C \subseteq \{0,1\}^m$ such that starting at any $x \in C$ if we do fewer than d corruptions to x, we do not end up at any $y \in C$ with $y \neq x$. When the S_i are all of size b and form a partition of $[m]$ into $t = m/b$ parts, then such a code is exactly the same as a classical Hamming distance code an alphabet of size 2^b. Error-correcting graph codes give a first step into the challenging setting where the S_i all pairwise intersect - here we have $m = \binom{n}{2}$, $t = n$, the S_i (which correspond to all edges incident on a given vertex) all have size $n-1$, and every pair S_i and S_j intersect in exactly 1 element. It would be interesting to develop this theory – to both find the limits of what is achievable and to develop techniques for constructing codes against this error model.

Finally, there are many other themes from classical coding theory that could make sense to study in the context of graph codes and graph distance, including in the context of sublinear time algorithms. It would be interesting to explore this.

Organization of this paper

We set up basic notions in Section 2. We show the existence of optimal graph codes in Section 3. In Section 4 we construct asymptotically good codes. Finally, in Section 5, we show explicit constructions of graph codes with very high distance.

2 Graph codes: Basics

Definitions and notations

All graphs will be simple, undirected graphs on the vertex set $[n]$ unless otherwise noted. For any graph G, use A_G to denote the adjacency matrix of G, and view A_G as an element of the vector space $\binom{n}{2}_2$. For two graphs G, H, let $G\Delta H$ be the symmetric difference of the two graphs, i.e. $A_{G\Delta H} = A_G - A_H$. For a subset $S \subseteq [n]$, we use $G[S]$ to denote the subgraph of G induced by the vertex set S. If A is a $n \times n$ matrix and $S, T \subset [n]$, let $A_{S,T}$ be the sub-matrix indexed by S on the rows and T on the columns. For $x \in [0,1]$, we use $h_2(x) = -x\log_2 x - (1-x)\log_2(1-x)$ to denote the binary entropy function.

▶ **Definition 1** (Graph distance and relative graph distance).

- *The* graph distance *between two graphs G and H, denoted by $\mathrm{d}_{\mathrm{graph}}(G,H)$, is the smallest $d \in \mathbb{N}$ such that there is a set $S \subseteq [n]$, $|S| = d$, and $G[[n] \setminus S] = H[[n] \setminus S]$.*
- *The* relative graph distance*, or simply relative distance, between G and H is denoted by $\delta_{graph}(G,H)$, and is the quantity $\frac{\mathrm{d}_{\mathrm{graph}}(G,H)}{n}$.*

In the above definition, we require that the graphs $G[\overline{S}]$ and $H[\overline{S}]$ be identical and not just isomorphic. Lemma 2 describes several equivalent characterizations of graph distance.

▶ **Proposition 2** (Alternate characterizations of $\mathrm{d}_{\mathrm{graph}}$). *Suppose G and H are graphs on the same vertex set. Then*

1. *$\mathrm{d}_{\mathrm{graph}}(G,H)$ is the minimum vertex cover size of $G\Delta H$.*
2. *$\mathrm{d}_{\mathrm{graph}}(G,H)$ is the minimum number of vertices whose neighborhoods you need to edit to transform G into H*
3. *$\mathrm{d}_{\mathrm{graph}}(G,H)$ is the minimum number of vertices whose neighborhoods you need to edit to transform $G\Delta H$ into the empty graph.*

Note that $\mathrm{d}_{\mathrm{graph}}$ is a metric (see [19], Lemma 5).

▶ **Definition 3** (Graph code). *We say that a set $C \subseteq 2^{\binom{[n]}{2}}$ is a* graph code *on $[n]$ with distance d if for every pair of graph $G, H \in C$, we have that $\mathrm{d}_{\mathrm{graph}}(G,H) \geq d$.*

- *The* rate *of C, denoted by R_C, is the quantity $\frac{\log_2(|C|)}{\binom{n}{2}} \geq \frac{2\log_2(|C|)}{n^2}$.*
- *The* distance *(resp.* relative distance*) of C, denoted by d_C (resp. δ_C), is the quantity $\min_{\substack{G,H\in C\\ G\neq H}} \mathrm{d}_{\mathrm{graph}}(G,H)$ (resp. $\min_{\substack{G,H\in C\\ G\neq H}} \delta_{graph}(G,H)$).*

Upper bound

As noted in [19], the Singleton bound can be used to obtain an upper bound on the rate of a graph code. We include a proof here for completeness.

▶ **Proposition 4.** *Any graph code with relative distance δ has dimension at most $\binom{n(1-\delta)+1}{2}_2$.*

Proof. Consider any graph code C of relative distance δ. Let $A \subset [n]$ be any subset of at most $\delta n - 1$ vertices. For any two distinct $G_1, G_2 \in C$, we have that the graphs induced on the vertices outside A, $G_1[[n] \setminus A]$ and $G_2[[n] \setminus A]$, are different. Indeed, since otherwise, A is a vertex cover of $G_1 \Delta G_2$, contradicting the relative distance assumption. So, we have that

$$|C| \leq 2^{\binom{n(1-\delta)+1}{2}}.$$ ◀

Expressed in terms of rate and distance, Proposition 4 implies that for constant relative distance δ, $R \leq (1-\delta)^2 + O(1/n)$.

3 Existence of optimal graph codes

As with other objects in the theory of error-correcting codes, the first question we seek to answer relates to the optimal rate-distance tradeoff.

In contrast to the Hamming world, we find that, in the graph distance, random linear codes meet the Singleton bound.

▶ **Proposition 5.** *Let $\delta \in (0,1)$. Then, there exists a linear graph code with distance greater than δ and dimension at least*

$$\max\left\{\binom{n(1-\delta)}{2} - h_2(\delta)n - 2, 0\right\}.$$

Proof. We only consider the case when $\binom{n(1-\delta)}{2} - h_2(\delta)n - 2 > 0$, and prove this by a probabilistic construction. Let $\mathbf{G}_{n,1/2}$, be the Erdös-Rényi random graph distribution where the vertices are $[n]$, and each of the $\binom{n}{2}$ possible edges are selected independently with probability $1/2$. Let $k = \binom{n(1-\delta)}{2} - h_2(\delta)n - 2$, and let $G_1, ..., G_k$ be graphs chosen independently from $\mathbf{G}_{n,1/2}$. Consider the $\mathbb{F}_2$-linear space $C = \text{span}\{A_{G_1}, \ldots, A_{G_k}\}$. We wish to show that C has distance at least δn with high probability.

For $\vec{\alpha} \in \mathbb{F}_2^k$, let $H_{\vec{\alpha}}$ be the graph with the adjacency matrix $\sum_{i=1}^{k} \alpha_i A_{G_i}$. Recalling the definition of graph distance, we need that for any distinct $\vec{\alpha}, \vec{\beta} \in \mathbb{F}_2^k$, $H_{\vec{\alpha}} \Delta H_{\vec{\beta}}$ must have minimum vertex cover size at greater than δn. Since $H_{\vec{\alpha}} \Delta H_{\vec{\beta}} = H_{\vec{\alpha}-\vec{\beta}}$, it suffices to show that for any non-zero $\vec{\alpha} \in \mathbb{F}_2^k$, that $H_{\vec{\alpha}}$ has no vertex cover of size δn.

For any $\vec{\alpha} \in \mathbb{F}_2^k \setminus \{\vec{0}\}$, $H_{\vec{\alpha}}$ has the same law as $\mathbf{G}_{n,\frac{1}{2}}$. Let $B_{\vec{\alpha}}$ be the event that $H_{\vec{\alpha}}$ has a vertex cover of size δn. We have

$$\begin{aligned}
\Pr(B_{\vec{\alpha}}) &= \Pr\left(\exists S \in \binom{[n]}{\delta n} : G[[n] \setminus S] = \text{ is the empty graph}\right) \\
&\leq \binom{n}{\delta n} \cdot 2^{-\binom{n(1-\delta)}{2}} \\
&\leq 2^{-\binom{n(1-\delta)}{2} + h_2(\delta)n},
\end{aligned}$$

where the first inequality uses the union bound over subsets of size δn, and the fact that there are $\binom{n(1-\delta)}{2}$ edges outside of a vertex cover that all have to be unselected. Then, union bounding over the choices of $\vec{\alpha}$, we get

$$\Pr\left(\bigcup_{\alpha \in \mathbb{F}_2^k \setminus \{\vec{0}\}} B_{\vec{\alpha}}\right) \leq 2^{k - \binom{n(1-\delta)}{2} + h_2(\delta)n} \leq 1/4.$$

Therefore, with probability at least $3/4$, there does not exist $\alpha \in \mathbb{F}_2^k$, such that $\alpha \neq \vec{0}$, and H_α has a vertex cover of size δn, and hence C is a code with relative distance greater than δ.

Finally, note that if H_α does not have a vertex cover of size δn, then H_α is not the empty graph. Thus, the fact that there does not exist $\alpha \in_2^k$, $\alpha \neq \vec{0}$ such that H_α has a vertex cover of size δn implies that $G_1, ..., G_k$ are linearly independent. Hence, the dimension of C is $k = \binom{n(1-\delta)}{2} - h_2(\delta)n - 2$, as required. ◀

As a result, we have the following corollaries.

▶ **Corollary 6.** *For any constant $\delta \in (0, 1)$, there exist optimal linear graph codes with relative distance at least δ.*

▶ **Corollary 7.** *For any constant $c > 2$, there exists a linear graph code with dimension at least $\Omega(\log^2 n)$ and relative distance at least $\delta = 1 - c \cdot \frac{\log n}{n}$.*

4 Explicit graph codes for high distance: Concatenated Codes

To get explicit codes of distance $\delta > 1/2$, we start with Construction 3 of [19]. The construction utilizes the tensor product code introduced by [17], where elements of the code are matrices where all rows and columns are codewords over some base Hamming code. The elements of the tensor code are then the adjacency matrices of graphs. Since we consider undirected graphs and do not allow self-loops, we take the subcode of the tensor code containing only symmetric matrices with zeros on the diagonal.

▶ **Definition 8** (Symmetric Tensor Code with Zeros on the Diagonal)**.** *Let C be a code over $_q$. The symmetric tensor code with zeros on the diagonal built on C denoted* $\mathrm{STCZD}(C)$ *is the set of matrices A over $_q^{n\times n}$ such that (1) A is symmetric, (2) the rows and columns of A are codewords of C, and (3) the entries on the diagonal are all 0.*

Properties of elements of Tensor Product Codes that are symmetric and zero-diagonal were also previously studied, in the context of constructing a gap-preserving reduction from SAT to the Minimum Distance of a Code problem, by Austrin and Khot [4].

We will also define another notion of distance that will be useful later on.

▶ **Definition 9** (Directed graph distance)**.** *Let A, and B be $n \times n$ matrices over some field. Define the* directed graph distance *denoted* $\mathrm{d}_{\mathrm{directed}}(G, H)$ *to be the minimum d such that there exists sets $S, T \subset [n]$ of size d where $(A - B)_{\overline{S},\overline{T}} = \mathbf{0}$.*

For weighted, directed graphs, G, and H, abbreviate $\mathrm{d}_{\mathrm{directed}}(A_G, A_H) = \mathrm{d}_{\mathrm{directed}}(G, H)$. To better distinguish between $\mathrm{d}_{\mathrm{directed}}$ and $\mathrm{d}_{\mathrm{graph}}$, we sometimes refer to $\mathrm{d}_{\mathrm{graph}}$ as the *undirected* graph distance.

When G and H are weighted directed graphs, $\mathrm{d}_{\mathrm{directed}}(A_G, A_H)$ can be viewed as the minimum d such that you can go from G to H by editing the incoming edges of d vertices and the outgoing edges of d vertices. The main difference between directed and undirected graph distance is that directed graph distance allows the subset of rows and the subset of columns to be edited to be different. Insisting that $S = T$ in the definition for directed graph distance recovers the undirected graph distance. From this, it easily follows that if G and H are undirected graphs, then $\mathrm{d}_{\mathrm{directed}}(G, H) \leq \mathrm{d}_{\mathrm{graph}}(G, H)$. Thus, to find codes with high graph distance, it suffices to find codes with large directed graph distance, where all the elements are adjacency matrices of undirected, unweighted graphs (i.e., 0/1 matrices that are symmetric and zero diagonal). Note that when discussing rate directed graph codes C, we are referring to the quantity $\log_q(|C|)/n^2$ instead of $\log_q(|C|)/\binom{n}{2}$.

In the next lemma, we show several properties of $\mathrm{STCZD}(C)$. Most importantly, the Hamming distance of C translates to the directed graph distance of $\mathrm{STCZD}(C)$.

▶ **Lemma 10.** *Let C be a linear $[n,k,d]_q$-code, then* $\mathrm{STCZD}(C) \subset_q^{n\times n}$ *is linear, has dimension at least* $\binom{k+1}{2} - n$*, and has directed graph distance d.*

Proof. Let C be a linear $[n,k,d]_q$-code, and let $C' = \mathrm{STCZD}(C)$. C' is linear because C is linear, and the sum of symmetric matrices is symmetric.

WLOG, we assume that C is systematic, i.e., it has $k \times n$ generator matrix $G = [I|A]$, where I is the $k \times k$ identity and A is a $k \times (n-k)$ matrix. Then, for every $X \in_q^{k\times k}$, the following has rows and columns belonging to C

$$G^T X G = \begin{bmatrix} X & XA \\ A^T X & A^T XA \end{bmatrix}.$$

Furthermore, $G^T XG$ is symmetric and has zeros on the diagonal iff X is symmetric, X has zeros on the diagonal, and $A^T XA$ has zeros on the diagonal. This imposes $\binom{k+1}{2} + (n-k)$ linear constraints on the entries of X. Thus, the subspace of X for which $G^T XG \in C'$ has dimension at least $k^2 - \binom{k+1}{2} - (n-k) = \binom{k+1}{2} - n$.

Since C' is linear, to show the distance property, it suffices to show that $\mathrm{d}_{\mathrm{directed}}(A, \mathbf{0}) \geq d$ for every non-zero $A \in C'$. Let $A \in C'$ be a non-zero element of C', we'll show that for any $S, T \subset [n]$, with $|S| < d$, and $|T| < d$, $A_{\overline{S},\overline{T}} \neq \mathbf{0}$.

Since A is non-zero, there is some non-zero entry A_{ij}. Since the rows are elements of a linear code of distance d, the Hamming weight of the ith row is at least d. Since $|T| < d$, there is some $j' \notin T$ such that $A_{ij'}$ is non-zero. Then, the j'th column is also a non-zero codeword of C, so it also has Hamming weight at least d. Since $|S| < d$, there is some $i' \notin S$ such that $A_{i'j'}$ is non-zero. Thus, $A_{\overline{S},\overline{T}} \neq \mathbf{0}$. ◀

▶ Remark 11. A simple calculation shows that if C has constant rate, R, then $\mathrm{STCZD}(C)$ has rate $R^2/2 - o(1)$ as a directed graph code.

Given this lemma (and using the fact that $\mathrm{d}_{\mathrm{graph}} \geq \mathrm{d}_{\mathrm{directed}}$), for any **binary** code $C \subset_2^n$ with rate R and relative distance δ, $\mathrm{STCZD}(C)$ is a (undirected) graph code with rate $R^2 - o(1)$, and relative distance δ. Thus, if C has rate distance tradeoff $R = f(\delta)$, then $\mathrm{STCZD}(C)$ has rate distance tradeoff $R = f(\delta)^2 - o(1)$. Immediately, we get that taking the STCZD of any asymptotically good binary code yields an asymptotically good graph code.

There are two problems with this construction. Firstly, these codes may not be strongly explicit. Secondly, the Plotkin bound [13] implies that any binary code with distance $> 1/2$ has vanishing rate. So this falls short of our goal of obtaining strongly explicit, asymptotically good codes with $\delta > 1/2$.

We will address the first problem by showing that if the base code is a Reed Solomon code [14], then there is a large subcode that is strongly explicit.

▶ **Code 12** (Reed Solomon Code $\mathrm{RS}(n,R,q)$ [14])**.** *The Reed Solomon Code with parameters n, R, and q, where $q \geq n$, is a code over $\mathbb{F}_q^n$ with rate R and distance $1-R$.*

▶ **Lemma 13.** *Let $C \in \mathrm{RS}(n,R,q)$ where $Rn = k-1$. Then, there exists a* strongly explicit *subcode $S \subset \mathrm{STCZD}(C)$ such that the dimension of S is at least $\binom{k-1}{2}$.*

Proof. Essentially, we will evaluate symmetric polynomials that are a multiple of $(X-Y)^2$ on a $n \times n$ grid.

Suppose $h(X,Y)$ is a symmetric polynomial of individual degree at most $k-3$, and let M be the evaluations of $f(X,Y) = (X-Y)^2 h(X,Y)$ on a $n \times n$ grid. M is symmetric and has zeros on the diagonal. Furthermore, for a fixed value, y, $f(X,y)$ is a univariate polynomial

in X of degree at most $k-1$, and hence the column indexed by y is an element of a Reed Solomon code of dimension k, and block length n. Similarly, the rows are also elements of the same code. Thus $M \in \text{STCZD}$.

Let S be the space of bivariate symmetric polynomials of degree at most $k-3$. For $a, b \in \mathbb{N}$, define polynomials $p_{a,b}(X,Y) = X^aY^b + X^bY^a$. Notice that $p_{a,b}$ is symmetric, and furthermore the set

$$\{p_{a,b} : 0 \le a < b \le k-3\} \cup \{X^iY^i : i \in \{0,1,...,k-3\}\},$$

is linearly independent. Thus $\dim(S) = \binom{k-2}{2} + k - 2 = \binom{k-1}{2}$, as desired. ◀

To extend this construction to the setting of $\delta > 1/2$, we use the concatenation paradigm from standard error-correcting code theory, initially introduced by Forney [7].

We will start with a code over a large alphabet and then concatenate with an inner code, which will be an optimal directed graph code.

▶ **Lemma 14.** *For any $\epsilon > 0$, and sufficiently large n, for any $k < \epsilon^2 n^2 - 2n$, there exists a linear directed graph code over $_2$ of dimension k and distance at least $(1-\epsilon)n$.*

The proof is standard and similar to that of Proposition 5. So we will omit it.

▶ **Code 15** (Optimal Directed Graph Code $\text{Opt}(\epsilon, n, k)$)**.** *Require $k < \epsilon^2 n^2 - 2n$. Refer to a code with the properties in Lemma 14 as $\text{Opt}(\epsilon, n, k)$.*

4.1 Symmetric concatenation

Since our inner code is not guaranteed to be symmetric, simply replacing each field element in the outer code with its encoding might result in an asymmetric matrix. To remedy this, we transpose the encoding for entries below the diagonal. This is made formal below.

▶ **Definition 16** (Symmetric Concatenation)**.** *Let q, Q be prime powers, and n, N be positive integers. Let $C_{out} \subset_Q^{N\times N}$ and $C_{in} \subset_q^{n\times n}$ such that $|C_{in}| = Q$. Define $C_{in} \circ C_{out} \subset_q^{nN\times nN}$ to be the code obtained by taking codewords of C_{out} and replacing each symbol of the outer alphabet with by their encodings under C_{in} if they lie above or on the diagonal, and with the transpose of their encodings if they lie below the diagonal.*

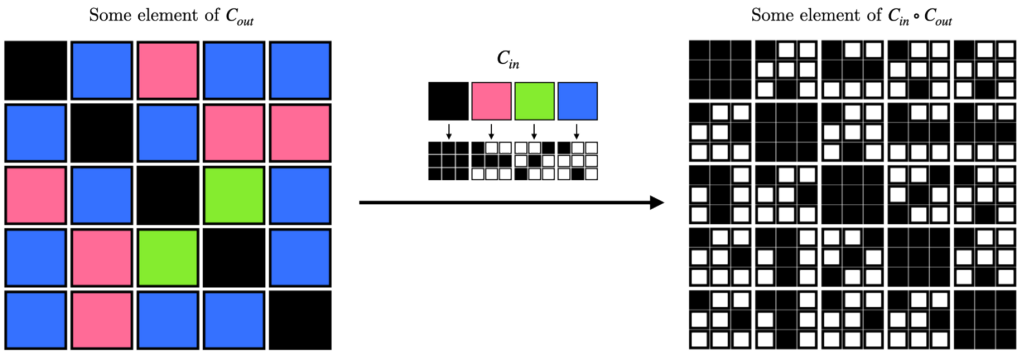


Figure 1 Example of symmetric concatenation. An outer codeword is shown on the left, with field elements represented as different colors. The concatenation with the inner code is shown to the right. Black squares represent 0, and white squares represent 1s.

Figure 1 visualizes an example of symmetric concatenation. We now show that distance and dimension concatenate exactly like it does for standard error-correcting codes.

▶ **Lemma 17.** *Suppose C_{in} and C_{out} are linear codes as in the previous definition with directed graph distance d and D, respectively. Let k be the dimension of C_{in}, and K be the dimension of C_{out}. Note $|C_{in}| = q^k = Q$. Then $C_{in} \circ C_{out}$ is linear and has distance at least dD, and dimension kK.*

Proof. Let $C = C_{in} \circ C_{out}$. First note that $C_{in} \circ C_{out}$ can be made linear by using a ${}_q$-linear map from ${}_{q^k}$ to ${}^k_q$ before encoding with the inner code.

Consider a non-zero outer codeword O, and let A be the codeword after concatenation. Let $S, T \subset [nN]$ be of size less than dD. We'll show that $A_{\overline{S},\overline{T}} \neq \mathbf{0}$. Partition A into $N \times N$ blocks, where the (I, J)'th block for $I, J \in [N]$, is the $n \times n$ matrix encoding the symbol at O_{IJ}. Identify the indices $[nN]$ with $[N] \times [n]$ where the tuple (I, i) corresponds to the i'th index in the I'th block.

For $I \in [N]$, let $S_I = \{i \in [n] : (I, i) \in S\}$ be the set rows in S in the I'th block. Define T_J similarly. Let $S_{\geq} = \{I \in [N] : |S_I| \geq d\}$, be the set of blocks in which there are at least d elements in S, and similarly define $T_{\geq}$, $S_{<}$, and $T_{<}$.

Since $\sum_{I \in [N]} |S_I| < dD$, $\sum_{J \in [N]} |T_J| < dD$, we have $|S_{\geq}| < D$, and $|T_{\geq}| < D$. Since the outer code has directed distance D, $O_{S_{<},T_{<}} \neq \mathbf{0}$, so there exists $I \in S_{<}$, and $J \in T_{<}$ such that $O_{I,J}$ is non-zero. So, the (I, J)'th block of A is a non-zero codeword or the transpose of a non-zero codeword of C_{in}. Let us call it X, and suppose that $X \in C_{in}$.

Since $|S_I| < d$, and $|T_J| < d$, and the inner code has distance at least d, we have that $X_{\overline{S_I},\overline{T_J}} \neq \mathbf{0}$. To finish the proof, note that $\mathrm{d}_{\mathrm{directed}}(X, \mathbf{0}) = \mathrm{d}_{\mathrm{directed}}(X^T, \mathbf{0})$ by switching the roles of S and T in the definition of directed graph distance.

For the claim about dimension, note that the number of codewords in C is the number of codewords in C_{out}, which is Q^K. The dimension of C is then $K \log_q(Q) = Kk$. ◀

Additionally, it is clear from the definition of symmetric concatenation that if C_{out} is symmetric and zero-diagonal, so is $C_{in} \circ C_{out}$.

▶ Remark 18. Lemma 17 also holds for the standard definition of concatenation (without transposing blocks below the diagonal). However, we will not need this fact.

4.2 Concatenated graph codes

We can instantiate the concatenated code using Reed Solomon codes.

▶ **Code 19** (Concatenated Code $C_{\mathrm{RS}}(\epsilon, n, k, N, \rho)$)**.** *Let $Q = 2^k$ be the size of the alphabet of the outer code. Let $\epsilon, \rho \in (0, 1)$, and n, k, N, to be integers satisfying $k < \epsilon^2 n^2 - 2n$, and $N \leq Q$. Then,*

$$C_{\mathrm{RS}} = \mathrm{Opt}(\epsilon, n, k) \circ \mathrm{STCZD}(\mathrm{RS}(N, \rho, Q)).$$

The following theorem follows directly from Lemmas 10 and 17. As a reminder, here we are considering the rate of the codes as a (undirected) graph code.

▶ **Theorem 20.** *Let ϵ, n, k, N, ρ be parameters satisfying the requirements listed in Code 19, then $C_{\mathrm{RS}}(\epsilon, n, k, N, \rho)$ is a graph code with rate $\epsilon^2\rho^2 - o(1)$, and relative distance $(1-\epsilon)(1-\rho)$.*

Note that using this construction, we can get asymptotically good codes for any constant rate and distance - including for distances $> 1/2$, which was not obtained in any of the previous constructions. We get $R = (1 - \sqrt{\delta})^4 - o(1)$ by setting $\epsilon = \rho$.

One drawback of this construction is that it is not strongly explicit or even explicit. The outer code can be made strongly explicit using Lemma 13, however, the inner code was an optimal code which we obtained by a randomized construction. The complexity of searching for such a code by brute force is too large. In particular, the optimal code has dimension $\epsilon^2 n^2$, and block length n^2. Since we need the size of the code to be equal to the size of the outer alphabet, we have $N = 2^{\epsilon^2 n^2}$, so $n = \sqrt{\log(N)}/\epsilon$. Then, there are at least $2^{\epsilon^2 n^4} = 2^{\log(N)^2/\epsilon^2}$ generator matrices to search over. Thus, we cannot find such a code efficiently.

To address this, we reduce the search space by concatenating *multiple times*. The resulting code will have a slightly worse distance/rate tradeoff but will still be asymptotically good for any constant distance or rate.

We also note that C_{RS} can also be made strongly explicit using a Justensen-like construction. However, although this code is again asymptotically good, it has distance bounded away from 1. We present this construction in the Appendix.

4.3 Multiple concatenation

While concatenating twice suffices to obtain an explicit code, it is not clear that the obtained code is strongly explicit. This may be addressed by concatenating three times, at the cost of slightly weaker parameters. Here, we will also use the tensor product code as a building block. For any linear code $C \subset_q^n$ let $\mathrm{TC}(C)$ be the tensor product code of C. As a reminder, $\mathrm{TC}(C)$ is the code consisting of matrices $A \in_q^{n \times n}$ such that each row and each column of A are elements of C.

▶ Remark 21. Suppose C is a linear code with distance d and rate R. Then, it follows from the proof of Lemma 10 that $\mathrm{TC}(C)$ has directed graph distance at least d. It is also well known that $\mathrm{TC}(C)$ has rate R^2.

Below we present the analysis for triple concatenation.

▶ **Code 22** (Triple Concatenation $\mathrm{C}_{\mathrm{Trip}}(\rho, N)$)**.** *For $\rho \in (0,1)$ and an integer N, let C be the subcode of $\mathrm{STCZD}(\mathrm{RS}(N, \rho, N))$ in Lemma 13. Then*

$$\mathrm{C}_{\mathrm{Trip}} = \mathrm{Opt}(N_3, \rho) \circ \mathrm{TC}(\mathrm{RS}(N_2, \rho, N_2)) \circ \mathrm{TC}(\mathrm{RS}(N_1, \rho, N_1)) \circ C,$$

where N_1, N_2 and N_3 are picked to make the concatenation work, i.e., $|\mathrm{Opt}(N_3, \rho)| \geq N_2$, and so on.

Notice that only the outer-most code needs to be symmetric and have zero diagonal since we use the symmetric concatenation operation (entries below the diagonal will be transposed). Thus, using the Tensor Product Code for the two middle codes (instead of STCZD) is sufficient and saves a factor of 2 (each time) on the rate.

▶ **Theorem 23.** *Let N be a positive integer and $\rho \in (0,1)$. Then $\mathrm{C}_{\mathrm{Trip}}(\rho, N)$ has distance at least $(1-\rho)^4$, and rate ρ^8. Furthermore, $\mathrm{C}_{\mathrm{Trip}}(\rho, N)$ is strongly explicit.*

Proof. The claims about rate and distance follow directly from Lemma 17.

We'll now show that this code is strongly explicit. The outermost code C is strongly explicit, and the two codes in the middle built on Reed Solomon codes are also strongly explicit. The idea is that the concatenation steps middle allow us to shrink the alphabet size from N to (less than) $\log(\log(N))$. Searching for optimal codes of this size can be done easily by brute force.

The dimension of $\mathrm{TC}(\mathrm{RS}(N_1, \rho, N_1))$ is $(\rho N_1)^2$, so the number of codewords is $N_1^{(\rho_1 N_1)^2}$, and for the concatenation to work, we need this to be at least N. That is, we need $(\rho_1 N_1)^2 \log(N_1) \geq \log(N)$, which we can get easily by setting $N_1 = O(\log(N))$. For the same reason, we can take $N_2 = O(\log\log(N))$.

This is now small enough to do a brute-force search for an optimal code. The inner-most code has dimension $\rho^2 N_3^2$, so we need $2^{\rho^2 N_3^2} = N_2$, or $N_3 = O(\sqrt{\log(N_2)})$. There are then $\rho^2 N_3^4$ possible generator matrices to search over. So the total number of codes we will need to search over is at most $2^{\rho^2 n^4} = 2^{O(\log\log\log(N)^2)} = 2^{o(\log\log(N))} = O(\log(N))$. ◀

The tradeoff for this code is then

$$R = (1 - \delta^{1/4})^8.$$

Thus, we get *strongly explicit* asymptotically good codes for any constant distance or rate.

If we just wanted explicit codes (instead of strongly explicit), concatenating twice would suffice. In particular, the search space for the inner-most code has size

$$2^{O(\log\log(N)^2)} = 2^{o(\log(N))},$$

which is smaller than any polynomial in N, but not polylogarithmic. The corresponding tradeoff for the double concatenated code is $R = (1 - \delta^{1/3})^6$.

5 Explicit graph codes with very high distance: dual-BCH Codes

In this section, we give explicit constructions of graph codes for the setting of very high distance ($\delta = 1 - o(1)$). As noted earlier, when the complete graph and the empty graph are part of the code, this is a generalization of the problem of constructing explicit Ramsey graphs (i.e., graphs with no large clique or independent set), which corresponds to graph codes of size at least 3.

Our main result here is an explicit construction of a graph code with distance $1 - \frac{n^\epsilon}{n^{1/2}}$ and dimension $\Omega(n^\epsilon \log n)$, for all $\epsilon \in [0, 1/2)$.

▶ **Theorem 24.** *For all d, there is a strongly explicit construction construction of a code with dimension $\Omega(d \log n)$ and distance $n - O(d\sqrt{n})$.*

In analogy with the situation for Hamming-distance codes, these are the dual-BCH codes of the graph-distance world.

5.1 Warmup: a graph code with dimension $\Omega(\log n)$

As a warmup, we first construct code with distance $1 - \frac{1}{n^\epsilon}$ with growing dimension.

Let $n = 2^t$. Let $\mathsf{Tr} :_{2^t} \to_2$ denote the finite field trace map. Concretely, it is given by:

$$\mathsf{Tr}(x) = x + x^2 + x^4 + \ldots + x^{2^i} + \ldots + x^{2^{t-1}}$$

For each $\alpha \in_{2^t}$, consider the matrix $M_\alpha \in_2^{n \times n}$, where the rows and columns of M_α are indexed by elements of $_{2^t}$, given by:

$$(M_\alpha)_{x,y} = \mathsf{Tr}(\alpha \cdot (x + y)^3).$$

Note that each M_α is symmetric. Consider the code

▶ **Code 25.** *For n of the form 2^t, let us define the family of codes*

$$C_{\text{Warmup}} = \{M_\alpha \mid \alpha \in_{2^t}\}.$$

We have that C_{Warmup} is a linear code of dimension $t = \log_2 n$.

▶ **Theorem 26.** *The distance of C_{Warmup} is at least $1 - O(n^{-1/2})$.*

Proof. Fix any $\alpha \in \mathbb{F}_{2^t}$. Let $S \subseteq_{2^t}$ be an arbitrary subset of vertices. It suffices to show that if S is bigger than $\Omega(n^{1/2}) = \Omega(2^{t/2})$, then there exist some $x, y \in S$ such that

$$\mathsf{Tr}(\alpha \cdot (x+y)^3) = 1.$$

Suppose not. Then we have:

$$\sum_{x,y \in S} (-1)^{\mathsf{Tr}(\alpha(x+y)^3)} = |S|^2.$$

By Cauchy-Schwarz, we get:

$$\begin{aligned}
|S|^4 &= \left(\sum_{x\in S}\sum_{y\in S}(-1)^{\mathsf{Tr}(\alpha(x+y)^3)}\right)^2 \\
&\le \left(\sum_{x\in S}\left(\sum_{y\in S}(-1)^{\mathsf{Tr}(\alpha(x+y)^3)}\right)^2\right)\cdot|S| \\
&\le \left(\sum_{x\in_{2^t}}\left(\sum_{y\in S}(-1)^{\mathsf{Tr}(\alpha(x+y)^3)}\right)^2\right)\cdot|S| \\
&= \left(\sum_{x\in_{2^t}}\sum_{y_1,y_2\in S}(-1)^{\mathsf{Tr}(\alpha((x+y_1)^3+(x+y_2)^3))}\right)\cdot|S| \\
&\le \left(\sum_{y_1,y_2}\left|\sum_{x\in_{2^t}}(-1)^{\mathsf{Tr}(\alpha((x+y_1)^3+(x+y_2)^3))}\right|\right)\cdot|S|.
\end{aligned}$$

For $y_1, y_2 \in_{2^t}$, let $P_{y_1,y_2}(X)$ be the polynomial

$$\begin{aligned}
P_{y_1,y_2}(X) &= \alpha\cdot\left((X+y_1)^3 + (X+y_2)^3\right) \\
&= \alpha\cdot\left((y_1+y_2)X^2 + (y_1^2+y_2^2)X + (y_1^3+y_2^3)\right).
\end{aligned}$$

The key observation is that for most $(y_1, y_2) \in S^2$, the trace of the polynomial $P_{y_1,y_2}(X)$ is a nonconstant ${}_2$-linear function, and thus the inner sum:

$$\sum_{x\in_{2^t}}(-1)^{\mathsf{Tr}(P_{y_1,y_2}(x))}$$

equals 0.

▶ **Lemma 27.** *If $P(X) = aX^2 + bX + c \in_{2^t} [X]$, then*

$$\mathsf{Tr}\circ P :_{2^t}\to_2$$

is a nonconstant ${}_2$-linear function unless $a = b^2$.

The proof is standard, and we omit it.

By the lemma, we get that there are at most $4|S|$ choices of (y_1, y_2) such that the inner sum is non-zero (namely those $(y_1, y_2) \in S^2$ for which $\alpha(y_1 + y_2) = \left(\alpha(y_1^2 + y_2^2)\right)^2$, which are few in number by the Schwartz-Zippel lemma).

Thus we get:

$$|S|^4 \leq 4|S|^2 \cdot 2^t,$$

from which we get $|S| \leq O(2^{t/2})$, as desired. ◀

5.2 Larger dimension

We now see how to get graph codes of distance $1 - \frac{1}{n^\epsilon}$ with $\epsilon < \frac{1}{2}$ and larger rate.

For a polynomial $f(X) \in_{2^t} [X]$, let M_f be $n \times n$ matrix with rows and columns indexed by $_{2^t}$ for which:

$$(M_f)_{x,y} = \mathsf{Tr}(f(x + y)).$$

Let W_d be the $_{2^t}$-linear space of all polynomials $f(X)$ of the form:

$$f(X) = \sum_{3 \leq 2i+1 \leq d} \alpha_i X^{2i+1},$$

where the $\alpha_i \in_{2^t}$.

Then, let us define our construction.

▶ **Code 28.** *For n of the form 2^t and $d \leq n^{1/2}$, let us define the family of codes*

$$C_{\text{DualBCH}}(n, d) = \{M_f : f \in W_d\}.$$

▶ **Theorem 29.** *We have that $C_{\text{DualBCH}}(n, d)$ is a linear graph code of distance $1 - O\left(dn^{-1/2}\right)$ and dimension $\Omega(dt) = \Omega(d \log n)$.*

Proof. The proof is very similar to the proof of Theorem 26. Consider any $M_f \in C_{\text{DualBCH}}(n, d)$ with $f \neq 0$. It suffices to show that the independence number[2] of M_f is $O(dn^{1/2})$.

Assume that $S \subseteq_{2^t}$ is an independent set in M_f. Then

$$|S|^2 = \sum_{x,y \in S} (-1)^{\mathsf{Tr}(f(x+y))}. \tag{3}$$

As in the proof of Theorem 26, by the Cauchy-Schwartz inequality and some simple manipulations, we get:

$$|S|^4 \leq \left(\sum_{y_1, y_2 \in S} \left| \sum_{x \in_{2^t}} (-1)^{\mathsf{Tr}(P_{y_1,y_2}(x))} \right| \right) \cdot |S|, \tag{4}$$

where:

$$P_{y_1,y_2}(X) = f(X + y_1) - f(X + y_2).$$

[2] An essentially identical proof shows that the clique number also has the same bound. The only change is to replace the LHS of (3) by $-(|S|^2 - |S|)$, and this sign change does not affect anything later because we immediately apply Cauchy-Schwarz to get (4).
This justifies our referring to this code as a "code of Ramsey graphs".

At this point, we need an upper bound in the inner sum:

$$U_{y_1,y_2} = \left| \sum_{x \in_{2^t}} (-1)^{\mathsf{Tr}(P_{y_1,y_2}(x))} \right|.$$

To get this, we will invoke the deep and powerful Weil bound:

▶ **Theorem 30** ([16], Chapter II, Theorem 2E)**.** *Suppose $P(X) \in_{2^t} (X)$ is a nonzero polynomial of odd degree with degree at most d. Then:*

$$\left| \sum_{x \in_{2^t}} (-1)^{\mathsf{Tr}(P(x))} \right| \leq O(d2^{t/2}).$$

We will use this to show that all but a few pairs $(y_1, y_2) \in S^2$, U_{y_1,y_2} are small.

▶ **Lemma 31.** *For all but $d|S|$ pairs $(y_1, y_2) \in S^2$,*

$$U_{y_1,y_2} \leq O(d2^{t/2}).$$

is at most $d|S|$.

Assuming this for the moment, we can proceed with Equation (4):

$$\begin{aligned} |S|^4 &\leq \left(d|S| \cdot 2^t + |S|^2 \cdot O(d \cdot 2^{t/2}) \right) \cdot |S| \\ &= d|S|^2 2^t + O(d|S|^3 2^{t/2}). \end{aligned}$$

Thus:

$$|S|^2 \leq d2^t + O(d|S|2^{t/2}),$$

which implies that $|S| \leq O\left(d \cdot 2^{t/2}\right)$, as desired. ◀

Proof of Lemma 31

Proof. Theorem 30 only applies to polynomials with odd degree. We first recall a standard trick involving the Tr map to deduce consequences for arbitrary degree polynomials.

Note that $\mathsf{Tr}(a^2) = \mathsf{Tr}(a)$ for all $a \in_{2^t}$, and that every element of $_{2^t}$ has a square root. Thus for any positive degree monomial $M(X) = \alpha X^i$, where $i = j \cdot 2^k$ with j odd, the equality:

$$\mathsf{Tr}(M(x)) = \mathsf{Tr}(\widetilde{M}(x))$$

for each $x \in_{2^t}$, where $\widetilde{M}(X)$ is the odd degree monomial given by:

$$\widetilde{M}(X) = \alpha^{1/2^k} X^j.$$

Extending by linearity, this allows us to associate, to every polynomial $P(X) \in_{2^t} [X]$, a polynomial $\widetilde{P}(X)$ with

$$\mathsf{Tr}(P(x)) = \mathsf{Tr}(\widetilde{P}(x))$$

for all $x \in_{2^t}$, and where every monomial of $\widetilde{P}(X)$ (except possibly the constant term) has odd degree.

The key observation is that whenever $\widetilde{P}_{y_1,y_2}(X)$ is nonconstant, it has odd degree, and so we can apply the Weil bound. In this case, since:

$$\mathsf{Tr}(P_{y_1,y_2}(x)) = \mathsf{Tr}(\widetilde{P}_{y_1,y_2}(x))$$

for each $x \in_{2^t}$, we get:

$$U_{y_1,y_2} = \left| \sum_{x \in_{2^t}} (-1)^{\mathsf{Tr}(\widetilde{P}_{y_1,y_2}(x))} \right| \tag{5}$$

$$\leq O(d \cdot 2^{t/2}), \tag{6}$$

where the last step follows from the Weil bound (Theorem 30).

Thus we simply need to show that there are at most $d|S|$ pairs $(y_1, y_2) \in S^2$ for which $\widetilde{P}_{y_1,y_2}(X)$ is a constant.

Suppose $f(X)$ has degree exactly $2e+1$. Let α be the coefficient of X^{2e+1} in $f(X)$.

Define $\gamma_i(Y) \in_q [Y]$ by:

$$f(X+Y) = \sum_{j=0}^{2e+1} \gamma_i(Y) X^i.$$

Note that $\deg(\gamma_i(Y)) \leq 2e+1-i$. It is easy to check that $\gamma_{2e+1}(Y) = \alpha$ and $\gamma_{2d}(Y) = \alpha Y$.

Then we have:

$$\begin{aligned} P_{y_1,y_2}(X) &= f(X+y_1) - f(X+y_2) \\ &= \sum_{i \leq 2e} \left(\gamma_i(y_1) - \gamma_i(y_2)\right) X^i. \end{aligned}$$

Then by definition,

$$\widetilde{P}_{y_1,y_2}(X) = \sum_{\substack{j \leq 2e-1 \\ j \text{ odd}}} \left(\sum_{\substack{k \geq 0 \\ j 2^k \leq 2e}} \left(\gamma_{j \cdot 2^k}(y_1) - \gamma_{j \cdot 2^k}(y_2)\right)^{\frac{1}{2^k}} \right) X^j.$$

We are trying to show that for most y_1, y_2, this is nonconstant. We will do this by identifying a monomial of positive degree which often has a nonzero coefficient. Let $e = j_0 \cdot 2^{k_0}$ with j odd. We will focus on the coefficient of X^{k_0}. It equals:

$$\left(\gamma_{2e}(y_1) - \gamma_{2e}(y_2)\right)^{1/2^{k_0+1}} + \left(\sum_{0 \leq k \leq k_0} \left(\gamma_{j \cdot 2^k}(y_1) - \gamma_{j \cdot 2^k}(y_2)\right)^{\frac{1}{2^k}} \right).$$

By linearity of the map $a \mapsto a^{1/2^k}$, this can be expressed in the form $Q(y_1^{1/2^{k_0+1}}, y_2^{1/2^{k_0+1}})$, where $Q(Z_1, Z_2)$ is a bivariate polynomial of degree at most $2e$. Furthermore, using the fact that $\gamma_{2e}(Y) = \alpha Y$, the homogeneous part of $Q(Z_1, Z_2)$ of degree 1 exactly equals:

$$\alpha^{1/2^{k_0+1}}(Z_1 - Z_2),$$

which is nonzero. Thus $Q(Z_1, Z_2)$ is a nonzero polynomial.

Thus by the Schwartz-Zippel lemma, for $T = \{y^{1/2^{k_0+1}} \mid y \in S\}$, there are at most $2e|T| \leq d|S|$ values of $(z_1, z_2) \in T^2$ such that $Q(z_1, z_2) = 0$. Thus there are at most $d|S|$ values of $(y_1, y_2) \in S^2$ for which the coefficient of X^{k_0} in $\widetilde{P}_{y_1,y_2}(X)$ is 0. Whenever it is nonzero, Equation (6) bounding U_{y_1,y_2} applies, and we get the desired result. ◀

References

1 Noga Alon. Connectivity graph-codes. *arXiv preprint*, 2023. arXiv:2308.07653.

2 Noga Alon. Graph-codes. *arXiv preprint*, 2023. arXiv:2301.13305.

3 Noga Alon, Anna Gujgiczer, János Körner, Aleksa Milojević, and Gábor Simonyi. Structured codes of graphs. *SIAM Journal on Discrete Mathematics*, 37(1):379–403, 2023. doi:10.1137/22M1487989.

4 Per Austrin and Subhash Khot. A Simple Deterministic Reduction for the Gap Minimum Distance of Code Problem. *IEEE Transactions on Information Theory*, 60(10):6636–6645, 2014. doi:10.1109/TIT.2014.2340869.

5 Shixi Chen and Shuigeng Zhou. Recursive mechanism: Towards node differential privacy and unrestricted joins. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 653–664. Association for Computing Machinery, 2013. doi:10.1145/2463676.2465304.

6 Pat Devlin. Personal Communication.

7 G David Forney. Concatenated codes. Technical report, Massachusetts Institute of Technology, Research Laboratory of Electronics, 1965.

8 Michael Hay, Chao Li, Gerome Miklau, and David Jensen. Accurate Estimation of the Degree Distribution of Private Networks. In *2009 Ninth IEEE International Conference on Data Mining*, pages 169–178, 2009. doi:10.1109/ICDM.2009.11.

9 Palak Jain, Adam Smith, and Connor Wagaman. Time-Aware Projections: Truly Node-Private Graph Statistics under Continual Observation. *arXiv*, 2024. doi:10.48550/arXiv.2403.04630.

10 J. Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18(5):652–656, 1972. doi:10.1109/TIT.1972.1054893.

11 Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Analyzing graphs with node differential privacy. In *Proceedings of the 10th Theory of Cryptography Conference on Theory of Cryptography*, TCC'13, pages 457–476. Springer-Verlag, 2013. doi:10.1007/978-3-642-36594-2_26.

12 James L Massey. Threshold decoding. Technical report, Massachusetts Institute of Technology, Research Laboratory of Electronics, 1963.

13 M. Plotkin. Binary codes with specified minimum distance. *IRE Transactions on Information Theory*, 6(4):445–450, 1960. doi:10.1109/TIT.1960.1057584.

14 I. S. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960. doi:10.1137/0108018.

15 Kai-Uwe Schmidt. Symmetric bilinear forms over finite fields with applications to coding theory. *Journal of Algebraic Combinatorics*, 42(2):635–670, 2015. doi:10.1007/s10801-015-0595-0.

16 W.M. Schmidt. *Equations over Finite Fields: An Elementary Approach*. Lecture Notes in Mathematics. Springer Berlin Heidelberg, 2006. URL: https://books.google.co.uk/books?id=up97CwAAQBAJ.

17 J. Wolf. On codes derivable from the tensor product of check matrices. *IEEE Transactions on Information Theory*, 11(2):281–284, 1965. doi:10.1109/TIT.1965.1053771.

18 Lev Yohananov, Yuval Efron, and Eitan Yaakobi. Double and Triple Node-Erasure-Correcting Codes Over Complete Graphs. *IEEE Transactions on Information Theory*, 66(7):4089–4103, 2020. doi:10.1109/TIT.2020.2971997.

19 Lev Yohananov and Eitan Yaakobi. Codes for Graph Erasures. *IEEE Transactions on Information Theory*, 65(9):5433–5453, 2019. doi:10.1109/TIT.2019.2910040.

A Justensen-like code

The construction in this example is inspired by the Justensen code [10], which uses an ensemble of codes for the inner code instead of a single inner code. Justensen uses an ensemble known as the Wozencraft Ensemble [12] with the following properties.

▶ **Theorem 32** (Wozencraft Ensemble). *For every large enough k, there exists codes $C^{(1)}, C^{(2)}, ..., C^{(2^k-1)}$ over $\frac{2}{2}^k$ with rate $1/2$, where $1-\epsilon$ fraction of them have distance at least $H_2^{-1}(1/2-\epsilon)$.*

Since our goal is graph distance, we use the STCZD operation to covert the Wozencraft Ensemble from codes over strings with good Hamming distance to codes of matrices with good graph distance.

▶ **Lemma 33** (Wozencraft Ensemble Modification). *For any $\epsilon > 0$, and large enough k, there exists codes $D^{(1)}, D^{(2)}, ..., D^{(2^k-1)}$ over $\frac{2k\times 2k}{2}$. View these as directed graph codes. Then these codes have rate $1/8$, and at least a $1-\epsilon$ fraction of them have distance at least $H_2^{-1}(1/2-\epsilon)$.*

Proof. Let $C^{(1)}, C^{(2)}, ..., C^{(N)}$ be the Wozencraft Ensemble. For each $I \in [N]$, define $D^{(I)} = \text{STCZD}(C^{(I)})$. Note that each $D^{(I)}$ is a code over $\frac{2k\times 2k}{2}$. Note that by lemma Lemma 10, each of the codes has rate $1/8$. Since the STCZD operation translates Hamming distance to directed graph distance, we also have the same guarantee as the original Wozencraft Ensemble - at least $(1-\epsilon)$ fraction of the codes have distance at least $H_2^{-1}(1/2-\epsilon)$. ◀

Concatenating STCZD(RS) with the modified Wozencraft Ensemble in a particular arrangement yields our next construction.

▶ **Code 34** (Justensen-like $C_{\text{Justensen}}(\epsilon, k, \rho)$). *Require $\epsilon, \rho \in (0,1)$. Let $Q = 2^k$, and $N = 2^k - 1$.*

Let $D^{(1)}, D^{(2)}, ..., D^{2^k-1}$ be the modified Wozencraft Ensemble Lemma 33.

Then $C_{\text{Justensen}}$ is the code where for each element of $A \in \text{STCZD}(\text{RS}(N, \rho, Q))$, for each $I, J \in [N]$, we replace the symbol at A_{IJ} with its encoding under $D^{(\min(I,J))}$. If $J < I$, we transpose the encoding (to keep the matrix symmetric).

Figure 2 shows where each inner code is applied.

$$\begin{bmatrix} - & D^{(1)} & D^{(1)} & D^{(1)} & D^{(1)} & D^{(1)} \\ D^{(1)T} & - & D^{(2)} & D^{(2)} & D^{(2)} & D^{(2)} \\ D^{(1)T} & D^{(2)T} & - & D^{(3)} & D^{(3)} & D^{(3)} \\ D^{(1)T} & D^{(2)T} & D^{(3)T} & - & D^{(4)} & D^{(4)} \\ D^{(1)T} & D^{(2)T} & D^{(3)T} & D^{(4)T} & - & D^{(5)} \\ D^{(1)T} & D^{(2)T} & D^{(3)T} & D^{(4)T} & D^{(5)T} & - \end{bmatrix}$$

Figure 2 Inner code arrangement for $C_{\text{Justensen}}$.

▶ **Theorem 35.** *For any $\epsilon, \rho \in (0,1)$, and k, a sufficiently large integer, $C_{\text{Justensen}}(\epsilon, \rho, k)$ is a strongly explicit linear graph code with rate $\rho^2/8 - o(1)$, and distance at least $(1-\rho-\epsilon)H^{-1}(1/2-\epsilon)$.*

Proof. Let $N = 2^k - 1$, and $n = 2k$ be the side lengths of the inner and outer codes, respectively. First note that $C_{\text{Justensen}}$ is a linear graph code over $\frac{nN\times nN}{2}$, since both the inner and outer codes are linear, and we can apply a $_2$ linear map from $_{2^k} \to \frac{k}{2}$ before encoding with the inner code.

Rate. By Lemma 10, the outer code, STCZD(RS(N, ρ, Q)), has rate $\rho^2/2 - o(1)$, and by Lemma 33, the inner codes have rate $1/8 - o(1)$. Thus, the rate is $\rho^2/8 - o(1)$ as an undirected graph code.

Distance. Let O be a non-zero outer codeword. For convenience, let $d = H^{-1}(1/2 - \epsilon)$, and $n = 2k$ be the side length of the inner code. We claim the distance is at least $(1 - \rho - \epsilon)d$.

Call $I \in [N]$, bad if the distance of $D^{(I)} < d$, and good otherwise. Let $B \subset [N]$ be the subset of bad indices. By the guarantee of the Wozencraft ensemble, we know that $|B| < \epsilon N$. Since O_{IJ} gets encoded with $\min(I, J)$, if $I, J \notin B$, then O_{IJ} is encoded with an inner code of distance at least d.

Define S_I, T_J as in the proof of Lemma 17. Let $S_{\geq} = \{I : |S_I| \geq d_{in} \text{ and } I \notin B\}$. Similarly, define $T_{\geq}$. Then $|S_{\geq}|, |T_{\geq}| < (1 - \rho - \epsilon)N$. Then $|S_{\geq} \cup B|, |T_{\geq} \cup B < (1 - \rho)N$. Since this is less than the outer distance of the code, we have that $O_{IJ} \neq 0$ for some $I \notin S_{\geq} \cup B$, and $J \notin T_{\geq} \cup B$. In other words, $|S_I| < d$, $|T_J| < d_{in}$, and O_{IJ} is encoded with a code of directed graph distance at least d. Thus, by the inner distance, there remains a non-zero element in the (I, J)th block outside of S_I, and T_J. ◀

New Pseudorandom Generators and Correlation Bounds Using Extractors

Vinayak M. Kumar
University of Texas at Austin, TX, USA

Abstract

We establish new correlation bounds and pseudorandom generators for a collection of computation models. These models are all natural generalization of structured low-degree $\mathbb{F}_2$-polynomials that we did not have correlation bounds for before. In particular:

- We construct a PRG for width-2 $\mathsf{poly}(n)$-length branching programs which read d bits at a time with seed length $2^{O(\sqrt{\log n})} \cdot d^2 \log^2(1/\varepsilon)$. This comes quadratically close to optimal dependence in d and $\log(1/\varepsilon)$. Improving the dependence on n would imply nontrivial PRGs for $\log n$-degree $\mathbb{F}_2$-polynomials. The previous PRG by Bogdanov, Dvir, Verbin, and Yehudayoff had an exponentially worse dependence on d with seed length of $O(d \log n + d2^d \log(1/\varepsilon))$.
- We provide the first nontrivial (and nearly optimal) correlation bounds and PRGs against size-$n^{\Omega(\log n)}$ AC^0 circuits with either $n^{.99}$ SYM gates (computing an arbitrary symmetric function) or $n^{.49}$ THR gates (computing an arbitrary linear threshold function). This is a generalization of sparse $\mathbb{F}_2$-polynomials, which can be simulated by an AC^0 circuit with one parity gate at the top. Previous work of Servedio and Tan only handled $n^{.49}$ SYM gates or $n^{.24}$ THR gates, and previous work of Lovett and Srinivasan only handled polynomial-size circuits.
- We give exponentially small correlation bounds against degree-$n^{O(1)}$ $\mathbb{F}_2$-polynomials which are set-multilinear over *some* arbitrary partition of the input into $n^{1-O(1)}$ parts (noting that at n parts, we recover *all* low degree polynomials). This vastly generalizes correlation bounds against degree-d polynomials which are set-multilinear over a fixed partition into d blocks, which were established by Bhrushundi, Harsha, Hatami, Kopparty, and Kumar.

The common technique behind all of these results is to fortify a hard function with the right type of extractor to obtain stronger correlation bounds for more general models of computation. Although this technique has been used in previous work, they rely on the model simplifying drastically under random restrictions. We view our results as a proof of concept that such fortification can be done even for classes that do not enjoy such behavior.

2012 ACM Subject Classification Theory of computation → Circuit complexity; Theory of computation → Pseudorandomness and derandomization

Keywords and phrases Pseudorandom Generators, Correlation Bounds, Constant-Depth Circuits

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.68

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2025/002/`

Funding *Vinayak M. Kumar*: Supported by NSF Grant CCF-2008076, CCF-2312573, and a Simons Investigator Award (#409864, David Zuckerman).

Acknowledgements We thank David Zuckerman for helpful discussions. We also thank anonymous reviewers for helpful comments. We thank Jeffrey Champion, Chin Ho Lee, and Geoffrey Mon for comments on an earlier draft of the paper.

1 Introduction/Outline of Results

Many central questions in complexity theory revolve around proving limitations of various computational models. For example, there are research programs which seek lower bounds against constant depth circuits, low degree polynomials over $\mathbb{F}_2$, and perhaps most famously the complexity class P.


© Vinayak M. Kumar;
licensed under Creative Commons License CC-BY 4.0



Usually, lower bounds against a simple class of n-bit Boolean functions $\mathcal{C}$ is established by demonstrating an explicit function f such that no $g \in \mathcal{C}$ can compute f on every input. This is referred to as *worst-case hardness*. However, we may not be satisfied with this in practice and stipulate that no $g \in \mathcal{C}$ can even *approximate* f. After all, if there exists a g that agrees with f on all but one point, the difference may be impossible to detect in practice. Furthermore, establishing average case hardness against $\mathcal{C}$ can allow us to create PRGs against $\mathcal{C}$ via the "hardness to randomness" framework introduced by Nisan and Wigderson [24], as well as show hardness results against related function classes, like the majority of functions in $\mathcal{C}$. This *average-case hardness* statement is exactly what the study of correlation bounds capture.

To formally define this, let D a distribution over $\{0,1\}^n$. Define the *correlation* of two Boolean functions $f, g : \{0,1\}^n \to \{0,1\}$ *over* D to be

$$\mathsf{corr}_D(f,g) = |\mathsf{E}_{x\sim D}[(-1)^{f(x)+g(x)}]|.$$

We will usually be concerned with $D = U_n$, the uniform distribution, and should be assumed so if no distribution D is specified. Notice that this quantity is a real number in $[0,1]$. For intuition, note that if $f = g$ or $f = 1 - g$, the correlation is 1, whereas if f and g only match on about half the inputs, the correlation becomes small. This fact allows us to observe correlation is the right notion, as $\mathsf{corr}(f,g)$ being small implies that g cannot predict f much better than a coin flip. For a function f and a function class $\mathcal{C}$, we can define $\mathsf{corr}(f,\mathcal{C}) = \max_{g\in\mathcal{C}} \mathsf{corr}(f,g)$. Hence the notion of f being average-case hard for $\mathcal{C}$ is captured by $\mathsf{corr}(f,\mathcal{C})$ being small.

In this paper, we are most interested in the case $\mathcal{C}$ is the class of low degree $\mathbb{F}_2[x_1,\ldots,x_n]$ polynomials. Establishing correlation bounds against low degree $\mathbb{F}_2$ polynomials is an extremely interesting and central question in complexity theory, as it is either necessary or sufficient to understand a plethora of other problems, some of which concern communication protocols, matrix rigidity, and PRGs for circuits. See Viola's survey [30] for a detailed exposition on this rich program.

Unfortunately, there is a "$\log n$-degree barrier" for PRGs and correlation bounds against low degree polynomials. Current PRGs and correlation bounds are asymptotically tight for constant degree polynomials, but become *trivial* at degree $\log n$ [29]. Getting nontrivial PRGs (or even correlation bounds) against $\log n$-degree polynomials has been a tantalizing and longstanding open problem.

Towards breaking this barrier, researchers have shown strong correlation bounds for *structured* subsets of low degree $\mathbb{F}_2$-polynomials (such as sparse polynomials [20, 26], tensors [3], small-read polynomials, and symmetric polynomials [4]) with the hope of being able to generalize them. In this work, we establish new correlation bounds and PRGs for computation models generalizing some of these polynomials, namely width-2 branching programs reading d bits at a time, AC^0 containing a small number of arbitrary symmetric or linear threshold gates, and set-multilinear polynomials.

Interestingly, all of these correlation bounds are obtained by taking a function hard for a more specific class of polynomials, and then *fortifying* it with a well suited extractor. Although such a fortification technique is not new and has been used for establishing stronger lower bounds for formulas [18, 8], they usually rely on the fact that upon randomly fixing a subset of variables of a formula, there are extremely few possibilities for the resulting function. Our work shows that extractor fortification is a much broader technique that can strengthen lower bounds against function classes even if they do not simplify greatly under a random restriction. In particular, our correlation bounds demonstrate extractor fortification can work if the function class, after a random restriction, has low communication complexity or good algebraic structure.

Inspired by this, we would like to show that extractors will always strengthen correlation bounds, no matter what the proof of the bound is. At a first glance, this may feel intuitive. However, due to technical reasons, this seems challenging to establish.

The remainder of this section is devoted to introducing and motivating each computational model studied, surveying prior work in the topic, and stating all key results proven.

1.1 Better Bounds and PRGs Against AC^0 with More $\{\mathsf{SYM}, \mathsf{THR}\}$ Gates

Our knowledge of hardness and PRG results for AC^0 is far more developed than that of TC^0. Our state of the art PRGs for AC^0 is Lyu's construction [22], which ε-fools polysize AC^0 circuits with seed length $\tilde{O}(\log^{d-1}(n)\log(n/\varepsilon))$, whereas the current best PRG of Hatami, Hoza, Tal, and Tell which (2^{-n^δ})-fools size- $O(n^{1+\delta})$ TC^0 circuits have seed length $O(n^{1-\delta})$ [15]. Due to this stark contrast in parameters, it is natural to gradually work upward from AC^0 by allotting a budget of SYM (calculates an arbitrary symmetric function) or THR (calculates an arbitrary linear threshold function) gates in the circuit. This approach has been explored for more than a decade [28, 20, 26], building upon the study of PRGs for $\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0$ circuits pioneered by Luby, Velicković, and Wigderson [21]. This context explains why this circuit class a compelling generalization of sparse polynomials (which can be written as a small-size parity of ands). All the mentioned works use the following function introduced by Razborov and Wigderson in 1993 [25] (all arithmetic is over $\mathbb{F}_2$).

$$\mathsf{RW}_{m,k,r}(x) = \sum_{i=1}^{m} \prod_{j=1}^{k} \sum_{\ell=1}^{r} x_{ij\ell} \tag{1}$$

Most recently, Servedio and Tan [26] use $\mathsf{RW}_{m,k,r}$ to uncorrelate against constant-depth size-$n^{O(\log n)}$ AC^0 circuits whose top gate is $\{\mathsf{SYM}, \mathsf{THR}\}$ (denoted as $\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0$). Their explicit bound is

$$\mathsf{corr}\left(\mathsf{RW}_{\sqrt{\frac{n}{\log n}}, \log n, \sqrt{\frac{n}{\log n}}},\ \{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0\right) \leq 2^{-\Omega(n^{.499})}.$$

Via techniques used in [20], this can be translated to correlation bounds against AC^0 circuits with up to $n^{.499}$ SYM gates or $n^{.249}$ THR gates. As can be surmised by the repeated occurrences of $n^{.499}$, the strength of the correlation bound dictates how many $\{\mathsf{SYM}, \mathsf{THR}\}$ gates we can afford in our budget.

We show that RW is just one of many functions from a general class of hard functions with small correlation against $\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0$ circuits. For functions $f : (\{0,1\}^r)^k \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}^r$, denote $f \circ g^k(x_1 \ldots, x_k) := f(g(x_1), \ldots g(x_k))$.

▶ **Theorem 1** (informal). *Let g be computable by a size $n^{O(\log n)}$ $\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0$ circuit. Let f be average-case hard against multiparty protocols*[1]*, and let* Ext *be a suitable extractor. Then*

$$\mathsf{corr}(f \circ \mathsf{Ext}^{\cdot 01 \log n}, g) \leq 2^{-\Omega(n^{.999})}.$$

To our knowledge, this theorem gives the first context where generically precomposing with an extractor boosts correlation bounds whose proof does not rely on simplification under random restriction (indeed parity does not simplify under restriction and is contained in

[1] the formal condition is any function with small "k-party norm" or "cube norm", but this is currently the only technique we know that establishes average case hardness against multiparty protocols.

Table 1 Correlation bounds against $\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0_d$ circuits and the PRGs that follow via the [24] framework. In all previous work, the "hard" function used was the RW function, which was first considered by Razborov and Wigderson [25]. Our work uses a better suited function. This table is an extension of the one found in [26].

	Circuit type	Circuit size S	Correlation bound	PRG seed length
[28]	$\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0$	$n^{c \log n}$	$n^{-c_d \log n}$	$2^{O(\sqrt{\log(S/\varepsilon)})}$
[20]	$\mathsf{SYM} \circ \mathsf{AC}^0$	$n^{c \log\log n}$	$\exp(-n^{0.999})$	$2^{O\left(\frac{\log S}{\log\log S}\right)} + (\log(1/\varepsilon))^{2.01}$
[20]	$\mathsf{THR} \circ \mathsf{AC}^0$	$n^{c \log\log n}$	$\exp(-n^{0.499})$	$2^{O\left(\frac{\log S}{\log\log S}\right)} + (\log(1/\varepsilon))^{4.01}$
[26]	$\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0_d$	$n^{c \log n}$	$\exp(-\Omega(n^{0.499}))$	$2^{O(\sqrt{\log S})} + (\log(1/\varepsilon))^{4.01}$
This work	$\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0$	$n^{c \log n}$	$\exp(-\Omega(n^{0.999}))$	$2^{O(\sqrt{\log S})} + (\log(1/\varepsilon))^{2.01}$

$\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0\}$). Previously, extractors have only been used to boost correlation bounds for classes that heavily simplify under random restriction [18, 8].[2] Our theorem states that extractors can still boost correlation bounds, even if they were proven using communication complexity rather than random restrictions.

Furthermore, our theorem distills out the reason why RW was so effective as a hard function. Quantitatively, we can instantiate the template with a suitable extractor to obtain a new hard function with nearly-optimal correlation bounds.

Due to our strengthened correlation bounds, we can now get correlation bounds and PRGs against size-$n^{O(\log n)}$ AC^0 circuits with up to $n^{.999}$ SYM gates or $n^{.499}$ THR gates. Prior to this, no nontrivial correlation bound or PRG was known to handle such large size *and* number of $\{\mathsf{SYM}, \mathsf{THR}\}$ gates ([20] could handle the same number of $\{\mathsf{SYM}, \mathsf{THR}\}$ gates but only for $n^{O(\log\log n)}$-size circuits, and [26] could handle the same size circuits, but only $n^{.499}$ SYM or $n^{.249}$ THR gates).

Even for $\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0$ circuits which have only one $\{\mathsf{SYM}, \mathsf{THR}\}$ gate, our correlation bounds yields improved PRGs whose seed length is $2^{O(\sqrt{\log S})} + (\log(1/\varepsilon))^{2.01}$, which has a better dependence on ε, than previous work (see Table 1). In fact, since the best correlation bound one can hope for is $2^{-\Omega(n)}$, this dependence is almost optimal under the Nisan-Wigderson framework, and an alternative approach is needed to reach the optimal dependence of $\log(1/\varepsilon)$. Since any $\log n$-degree $\mathbb{F}_2$ polynomial can be expressed as a $\mathsf{SYM} \circ \mathsf{AND}_{\log n}$ circuit of size $n^{\log n}$, any improvement of the dependence of the seed length on S would give nontrivial PRGs for $\log n$-degree polynomials, a breakthrough result.

1.2 Much Better PRGs Against Width-2 Branching Programs Reading d Bits at a Time

Usually, one constructs PRGs for natural *computational* models, with the idea that we can drastically reduce the randomness we use if the randomized algorithm we are running can be simulated by such a model. Low degree polynomials is an extremely natural mathematical model with applications to circuit complexity, but some may not believe it is well grounded as

[2] There have been uses of extractors as a hard function against classes that do not simplify under restriction, like DNFs of Parities [9] and strongly read-once linear branching programs [12, 19, 7]. However they directly establish a correlation bound against the extractor rather than amplify a weaker hard function by precomposing with an extractor.

a computational one and thus not worth finding a PRG for. However, the work of Bogdanov, Dvir, Verbin, and Yehudayoff [5] showed the beautiful connection that PRGs for degree d polynomials are also PRGs against a particular model described as *width-2 length-*$\mathsf{poly}(n)$ *branching programs which read* d *bits at a time.*

▶ **Definition 2** ((d, ℓ, n)-2BP ([5], adapted)). *A (d, ℓ, n)-2BP (or more colloquially a width-2 length-ℓ branching program over n bits which reads d bits at a time) is a layered directed acyclic graph, where there are ℓ layers and each layer contains two nodes, which we label by 0 and 1. Each vertex in each layer j is associated with an arbitrary d-bit substring $x|_v$ of the input x. Each node in layer j has 2^d outgoing edges to layer $j+1$ that are labelled by all possible values in $\{0,1\}^d$. On input x, the computation starts with the first node v_{start} in the first layer, then follows the edge labelled by $x|_{v_{start}}$ onto the second layer, and so on until a node in the last layer is reached. The identity of this last node is the outcome of the computation.*

Such branching programs are a well motivated computation model which cover computation with only one bit of usable memory, low degree polynomials, and small width DNFs. The survey of unconditional PRGs by Hatami and Hoza refer to this model as a compelling computational model that places low degree polynomials in the computational landscape [14].

Unfortunately, there is a "$\log n$-degree barrier" for PRGs and correlation bounds against low degree polynomials. Current PRGs and correlation bounds are asymptotically tight for constant degree polynomials, but become *trivial* at degree $\log n$, as can be seen by the current best known PRG for degree-d polynomials by Viola which has seed length $O(d \log n + d2^d \log(n/\varepsilon))$ [29]. Getting nontrivial PRGs (or even correlation bounds) against $\log n$-degree polynomials has been a tantalizing and longstanding open problem, and thus PRGs for $(d, \mathsf{poly}(n), n)$-2BPs also seemingly appeared to inherit this "$d = \log n$ barrier" due to the reduction result of [5].

In this work, we construct PRGs against $(d, \mathsf{poly}(n), n)$-2BPs with exponentially better seed length, thereby giving nontrivial PRGs even in the regime $d = n^{1-o(1)}$. Define a d-junta to be a function $\phi : \{0,1\}^n \to \{0,1\}$ which is solely dependent on d input bits (i.e. can be written as $\phi'(x_i)_{i \in S}$ for some subset $S \subset [n]$ of size d). To get our shortened seed length, we evade the $\log n$-degree barrier by instead showing the equivalence between PRGs for $(d, \mathsf{poly}(n), n)$-2BPs and PRGs for the XOR of $\mathsf{poly}(n)$ many d-juntas (denoted as $\mathsf{JUNTA}_{n,d}^{\oplus \mathsf{poly}(n)}$). This class is already interesting in its own right, as it can be seen as a generalization of sparse $\mathbb{F}_2$-polynomials and combinatorial checkerboards (defined by Watson [32] and also studied by Gopalan, Meka, Reingold and Zuckerman [11]), as well as a specific class bounded collusion protocols studied by Chattopadhyay et al. [6]. However, we are not aware of any literature studying $\mathsf{JUNTA}_{n,d}^{\oplus m}$ specifically.

Our main technical contribution is strong correlation bounds for $\mathsf{JUNTA}_{n,d}^{\oplus \mathsf{poly}(n)}$. In particular, we show the following.

▶ **Theorem 3.** *There exists an explicit function f such that*

$$\mathsf{corr}(f, \mathsf{JUNTA}_{n,d}^{\oplus \mathsf{poly}(n)}) \leq \exp\left(-\frac{n}{d2^{O(\sqrt{\log n})}}\right)$$

By combining this with the "hardness-to-randomness" framework of Nisan and Wigderson [24], we construct a PRG of seed length $2^{O(\sqrt{\log n})} d^2 \log^2(1/\varepsilon)$. This is only a quadratic factor away from optimal dependence on d and ε. Improving the dependence on n would be a breakthrough, since if we set $n' = 2^{\sqrt{\log n}}$, a $(d, n, \mathsf{poly}(n))$-2BP can simulate any $\log n'$-degree polynomial over $x_1, \ldots x_{n'}$, and so having seed length $o(n')$ would effectively be breaking the $\log n$-degree barrier for $\mathbb{F}_2$-polynomial PRGs.

Interestingly enough, by combining an "simplification under restriction" approach pioneered by Ajtai and Wigderson [1] with a PRG for sparse $\mathbb{F}_2$-polynomials by Servedio and Tan [27], we are able to construct a PRG against $\mathsf{JUNTA}_{n,d}^{\oplus \mathsf{poly}(n)}$, and thus $(d, \mathsf{poly}(n), n)$-2BPs, with seed length $d2^{O(\sqrt{\log(n/\varepsilon)})}$. This gives us an optimal dependence on d, but an exponentially worse dependence on ε. This suggests perhaps with a combination of these two approaches, one might be able to achieve seed length $2^{O(\sqrt{\log n})} d \log(1/\varepsilon)$.

1.3 Near-Optimal Bounds Against High Degree Set-Multilinear Polynomials

As explained earlier, a central open question in complexity theory is to establish better-than-$O(1/\sqrt{n})$ nontrivial correlation bounds against $\Omega(\log n)$-degree polynomials. In order to make progress on this question, it is natural to consider structured low-degree $\mathbb{F}_2$-polynomials. This is what the work of Bhrushundi et al. does [3].

Define a polynomial $p : \{0,1\}^n \to \{0,1\}$ to be set-multilinear over a partition $X = (X_1, \ldots, X_d)$ of the input bits if every monomial contains *at most one* variable from each X_i (this is slightly more general than the usual definition of *exactly one*). The work of Bhrushundi et al. [3] prove that a random degree d set-multilinear tensor has exponentially small correlation against generic degree $d/2$ $\mathbb{F}_2$-polynomials for $d = \Omega(n)$. Towards making this correlation bound explicit, they defined $\mathsf{FFM}(X_1, \ldots, X_d) = \mathsf{lsb}(X_1 \cdot X_2 \cdots X_d)$, where multiplication is done by treating the X_i as field elements, and lsb outputs the least significant bit of the string. Bhushrundi et al. were able to give exponentially small correlation bounds against polynomials up to degree $o(n/\log n)$ which are set-multilinear over the fixed partition $(X_1, \ldots, X_d)$. However, this leaves more to be wanted. The partition with respect to which the polynomial is set-multilinear over needing to be fixed and dependent on FFM_d feels like an extremely strong and asymmetric condition. Can we uncorrelate against degree $< d$ polynomials set-multilinear over *any* equipartition of X into d parts? Can the parts be unequal? Can we have more than d of them?

We show the affirmative to all the above questions. If we take $\delta > 0$ to be an arbitrarily small constant, we can obtain exponentially small correlation against degree $< n^\delta$ polynomial for which there exists *some* partition of X into up to $n^{1-\delta}$ (not necessarily equal) parts such that p is set-multilinear over it. Notice improving $n^{1-\delta}$ parts to n would be a breakthrough, since all polynomials are set-multilinear over the n-partition of $X = (x_1, \ldots, x_n)$.

To do so, we fortify the hard function FFM with an extractor. Let $\mathsf{Ext}(X, W)$ be a strong linear seeded extractor (for each fixing of W, $\mathsf{Ext}(\cdot, W)$ is linear). For some parameter d, define the function

$$\mathsf{ExtFFM}_d(X_1, \ldots, X_d, W) := \mathsf{lsb}(\mathsf{Ext}(X_1, W) \cdot \mathsf{Ext}(X_2, W) \cdot \ldots \cdot \mathsf{Ext}(X_d, W)),$$

where multiplication is done over a finite field, and lsb outputs the least significant bit of the string. First note that $\mathsf{ExtFFM}_d(X_1, \ldots, X_d, W)$, for a fixed W, is set-multilinear over $X_1, \ldots, X_d$. Hence our intuition that set-multilinear polynomials might correlate the most with the hard function is preserved in ExtFFM as well. Using ExtFFM, we are able to obtain correlation bounds against the more intuitive notion of set-multilinear polynomials, where the structure of the partition does not matter. This gives more leeway since now if we want to implement this approach towards correlation bounds against low-degree polynomials, there is a larger class of set-multilinear polynomials that we can reduce generic polynomials to.

2 Technical Overview Of the Results

In this section, we give the overview of the proofs of the main results we covered above.

2.1 Stronger Correlation Bounds Against $\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0$

We focus on showing stronger correlation bounds against $\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0$, since the subsequent arguments turning this into PRGs against AC^0 with a few $\{\mathsf{SYM}, \mathsf{THR}\}$ gates are standard. The blueprint behind this argument follows the "simplification under restrictions" approach of previous works, but most similarly of Tan and Servedio [26]. A random restriction is a random partial assignment where for each variable, it is left unfixed (or "alive") with probability p, and is otherwise set to a uniform bit. [26] shows that under a random restriction, the hard function $\mathsf{RW}_{m,k,r}$ maintains integrity and uncorrelates against multiparty protocols, while the $\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0$ simplifies to a short multiparty protocol. However, the roadblock met in [26] preventing a correlation bound of $2^{-\Omega(n)}$ and only giving one of size $2^{-\Omega(n^{.499})}$ is due to parameters in $\mathsf{RW}_{m,k,r}$ being in contention with each other. To elucidate, if n is the input size, then we must have $mkr = n$. Via the analysis done in [26], the correlation bound ends up being in the form of $2^{-\Omega(m)} + 2^{-\tilde{\Omega}(r)}$, which forces any established correlation bound to be at best $2^{-\Omega(\sqrt{n})}$.

To understand why both conflicting terms show up, we give a quick overview of the argument of [26]. First, $\mathsf{RW}_{m,k,r}$ (as defined in Equation (1)) can be thought of as a fortified version of the *generalized inner product*, $\mathsf{GIP}_{m,k}(x_1, \ldots, x_k) := \sum_{i=1}^{m} \prod_{j=1}^{k} x_{ij}$, where each variable is now replaced by the parity of r new variables. This is effective against random restrictions, since as long as one of the r copies $x_{ij1}, \ldots, x_{ijr}$ survive the restriction, the corresponding term x_{ij} in GIP will survive. They argue that after a random restriction ρ is applied, the $\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0$ circuit simplifies to a short multiparty protocol, while $\mathsf{RW}_{m,k,r}|_\rho$ is still capable of computing $\mathsf{GIP}_{m/2,k}$ with high probability. Conditioning upon this, previous results of Babai, Nisan, and Szegedy [2] show that $\mathsf{GIP}_{m/2,k}$ has $2^{-\Omega(m/2^k)}$ correlation against these multiparty protocols, explaining the emergence of the $2^{-\Omega(m)}$ term in the correlation. Conditioning on $\mathsf{RW}_{m,k,r}|_\rho$ being able to compute $\mathsf{GIP}_{m/2,k}$ introduces an additive error to the correlation corresponding to the probability $\mathsf{RW}_{m,k,r}|_\rho$ *fails* to simplify. [26] bounds this by the chance all r copies of some variable x_{ij} becomes fixed by a restriction, which will be $(1-p)^r \approx \exp(-pr)$, explaining the occurrence of the $2^{-\Omega(r)}$ term in the correlation.

In summary, the argument of [26] requires r needs to be large to strongly fortify the hard function against random restrictions, while m needs to be large to have a stronger correlation bound against multiparty protocols. However, with the constraint $mr \leq n$, we are forced to compromise and reach the setting $m = r \approx \sqrt{n}$.

We now propose an abstraction of the hard function, which naturally yields a stronger correlation bound. If we define $\oplus_{m,r} : (\{0,1\}^r)^m \to \{0,1\}^m$ to be

$$\oplus_{m,r}(x_1, \ldots, x_m) = \left(\sum_{i=1}^{r} x_{1i}, \ldots, \sum_{i=1}^{r} x_{mi} \right),$$

we observe $\mathsf{RW}_{m,k,r} = \mathsf{GIP}_{m,k} \circ \oplus_{m,r}^k(x_1, \ldots, x_k) := \mathsf{GIP}_{m,k}(\oplus_{m,r}(x_1), \ldots, \oplus_{m,r}(x_k))$. The key insight is that our argument can be generalized to not just RW, but any function

$$f \circ \mathsf{Ext}^k := f(\mathsf{Ext}(x_1), \ldots, \mathsf{Ext}(x_k))$$

where f is average-case hard for multiparty protocols, and Ext is an *oblivious bit-fixing source extractor* (OBF extractor). Informally, an oblivious bit-fixing source extractor for min-entropy k is a function Ext such that if $\mathbf{X}$ is uniform over $\{0,1\}^n$ and ρ is a restriction which leaves $\geq k$ bits alive, the output $\mathsf{Ext}(\mathbf{X}|_\rho)$ is close to uniform. Recall our approach first applies a random restriction to simplify our circuit to a small multiparty protocol, which we then deal with using GIP. If the random restriction leaves sufficiently many variables alive with high probability, then $f \circ \mathsf{Ext}^k$ should still behave like f due to Ext being an OBF extractor. Since the circuit is now a multiparty protocol, the average-case hardness of f gives us a correlation bound.

Notice in the RW construction and the setting of parameters $m = r \approx \sqrt{n}$, $\oplus_{m,r}$ is an OBF extractor which maps n bits to $\sqrt{n}$ bits. But this means the input to the outer GIP function will only have $\approx \sqrt{n}$ bits, and so the best correlation bound we can hope to achieve is $\exp(-\Omega(\sqrt{n}))$. The restrictions used in the proof leave $n^{.99}$ variables alive with high probability, so intuitively we could hope that all these $n^{.99}$ "bits of randomness" could be preserved for GIP (or in general any f) rather than only $\sqrt{n}$, potetially resulting in a $\exp(-\Omega(n^{.99}))$ correlation bound alive. We do just this by using a much better OBF extractor of Kamp and Zuckerman [17]. By making this intuition more formal using techniques developed by Viola and Wigderson [31], we obtain $2^{-\Omega(n^{1-O(1)})}$ correlation bound. The idea of replacing parities with better suited extractors has also appeared in previous work [18, 8].

2.2 PRGs for $\mathsf{JUNTA}^{\oplus t}_{n,d}$ and (d,t,n)-2BPs

Our PRG construction blueprint can be briefly described as follows. We first establish correlation bounds against $\mathsf{JUNTA}^{\oplus t}_{n,d}$. We then put this through the Nisan-Wigderson "hardness vs. randomness" framework to create a PRG against $\mathsf{JUNTA}^{\oplus t}_{n,d}$. We then show that PRGs which fool $\mathsf{JUNTA}^{\oplus t}_{n,d}$ actually fool (d,t,n)-2BPs, making the $\mathsf{JUNTA}^{\oplus t}_{n,d}$ PRG our final construction. We first discuss why PRGs for $\mathsf{JUNTA}^{\oplus t}_{n,d}$ imply PRGs for (d,t,n)-2BPs, and then discuss the techniques needed to show strong correlation bounds against $\mathsf{JUNTA}^{\oplus t}_{n,d}$.

2.2.1 PRGs for $\mathsf{JUNTA}^{\oplus t}_{n,d} \implies$ PRGs for (d,t,n)-2BPs

Adopting the exposition in [14], the previous work of [5] can be outlined as follows. Consider a (d,t,n)-2BP B. By noticing that all transition functions in B are d-juntas, one can derive that $B(x) = B'(\phi_1, \ldots, \phi_{2t}(x))$, where B' is a $(1,t,2t)$ branching program. By Fourier expanding B', this can be decomposed as

$$B(x) = \sum_{S \subset [2t]} \widehat{B}(S)(-1)^{\sum_{i \in S} \phi_i(x)}.$$

[5] shows that $\sum_{S \subset [t]} |\widehat{B}(S)|$ is bounded , so by linearity of expectation and the Triangle Inequality, it suffices to fool the terms $(-1)^{\sum_{i \in S} \phi_i(x)}$. The approach in [5] makes the observation that each ϕ_i, by virtue of being a d-junta, can be written as a degree d polynomial. Consequently, a PRG for degree d polynomials will fool (d,t,n)-2BPs with seed length $O(d \log n + d2^d \log(n/\varepsilon))$. The issue here is that at $d = \log n$, the seed length becomes trivial.

However, we can notice that the $\mathbb{F}_2$-polynomial $p(x) \coloneqq \sum_{i \in S} \phi_i(x)$ has some additional structure. If $t = \mathsf{poly}(n)$, p is the sum of only a polynomial number of d-juntas. If there was a way to leverage this, and get a better PRG that fools $\mathsf{JUNTA}^{\oplus t}_{n,d}$, then we might hope to get nontrivial PRGs even in the regime $d = \Omega(\log n)$.

This observation already yields nontrivial PRGs for $d = \omega(\log n)$. Servedio and Tan [26] provide a PRG fooling $\mathbb{F}_2$-polynomials with S terms with seed length $2^{O(\sqrt{\log S}}\log(1/\varepsilon))$. Since each junta can be written as a polynomial with up to 2^d terms, each $g \in \mathsf{JUNTA}_{n,d}^{\mathsf{poly}(n)}$ can be written as a polynomial with $S = 2^d\mathsf{poly}(n)$ terms, yielding a PRG with seed length $(2^{O(\sqrt{d})}+O(\log n))\log(1/\varepsilon)$. Hence we get nontrivial seed length for $d = o(\log^2 n)$. [3] However, we proceed alternatively to get an exponentially better seed length.

2.3 The Nisan-Wigderson Framework and Correlation Bounds for $\mathsf{JUNTA}_{n,d}^{\oplus\mathsf{poly}(n)}$

We will once again use $f \circ \mathsf{Ext}^k$ as our hard function[4] to establish exponentially small correlation bounds against the class, and then apply the Nisan-Wigderson [24] framework to construct the PRG. The latter portion is straightforward, so we focus on establishing the correlation bounds.

Let $g \in \mathsf{JUNTA}_{n,d}^{\oplus\mathsf{poly}(n)}$. We first show that there exist a subset of variables, S, such that upon arbitrarily fixing bits outside of this set, g can be expressed as a sparse $\mathbb{F}_2$ polynomial, whereas each input block of $f \circ \mathsf{Ext}^k$ heavily intersect S. Hence if we fix $X_{\bar{S}}$ and take the correlation over S, each input block still maintains high min-entropy while g becomes a sparse polynomial, which is a small $\mathsf{SYM} \circ \mathsf{AC}^0$ circuit. Since the hard function is also the same, we can then apply techniques in the previous section to conclude.

2.4 Correlation Bounds against Set-Multilinear Polynomials

Recall that [3] has shown FFM_d uncorrelates against any lower degree polynomial which is set-multilinear over $(X_1, \ldots, X_d)$. The key ingredient behind proving strong correlation bounds against set-multilinear polynomials over arbitrary parititons is to first fortify each input block with extractors, and instead consider ExtFFM_d. This allows us to establish the following structural lemma, which intuitively states that even if you do not start out with a polynomial that is set-multilinear over $(X_1, \ldots, X_d)$, if not too many bits in each input block can be restricted to 1s such that the resulting function is set-multilinear over $(X_1, \ldots, X_d)$ induced by the live variables in each block, exponential correlation bounds can still be obtained.

▶ **Theorem 4.** *Let g be a polynomial of degree $< d$. Let $S_1, \ldots, S_d \subset [n/d]$ be subsets, and let ρ denote the restriction created by fixing the bits in X_i whose index is outside S_i to 1 for each $i \in [d]$. If the restricted function $g|_\rho(X_1, \ldots, X_d)$ becomes set-multilinear in $(X_1, \ldots, X_d)$, then have*

$$\mathsf{corr}(\mathsf{ExtFFM}_d, g) \leq 2^{-\Omega(\frac{n}{cd})}.$$

To explain the proof at a high level, if the sets S_i we leave alive aren't too small, then our strong extractor (conditioned on a good seed) will keep each block $\mathsf{Ext}(X_i, W)$ approximately uniform, and since the restricted function $g|_\rho$ is now set-multilinear over $(X_1, \ldots, X_d)$ we may use a similar approach as [3] to prove the theorem.

It turns out that via a combinatorial argument, one can show that polynomials which are set-multilinear over a large number of blocks can be turned into polynomials set-multilinear over $(X_1, \ldots, X_d)$ by fixing not too many bits per input block X_i. The correlation bounds then follow by the structural lemma.

[3] it is actually the case that a PRG from [21] already gets nontrivial seed length in the same regime, albeit with exponentially worse dependence in ε

[4] we also precompose with parities in the formal argument

3 Preliminaries

For positive integer n, $[n] := \{1, \ldots, n\}$ and $\binom{[n]}{s}$ is the set of all subsets of $[n]$ with $|S| = s$. We denote $e(x) := (-1)^x$.

3.1 Convention About Input Blocks

We will canonically fix a partition of bit strings into d contiguous blocks, each with n/d bits. In particular, any $X \in \{0,1\}^n$ can be written as $X = (X_1, \ldots, X_d)$ where each X_i is the n/d-bit substring. If a string $Y \in \{0,1\}$ is defined, Y_i will be assumed to mean the length n/d substring of Y contained in the ith input block, defined with respect to the canonical partition. Also, we will denote $X_{-i} := (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_d)$ to be the input with the ith block removed.

For a string $X \in \{0,1\}$, we may sometimes identify the n/d bit string X_i as an n-bit string in the following way: the ith block is filled with X, and all other blocks are filled with 0s. Hence, if we interpret bit strings as elements of $\mathbb{F}_2^n$, and we have $X, Y \in \mathbb{F}_2^n$, the expression $X + Y_i$ is well defined.

For parameters $k, d \leq n$ and two functions $f : (\{0,1\}^m)^k \to \{0,1\}$ and $g : \{0,1\}^{n/k} \to \{0,1\}^d$, we will define

$$f \circ g^k = f(g(X_1), \ldots, g(X_d)).$$

3.2 Finite Fields

We will be working with finite fields of characteristic 2. For the finite field over 2^n elements, $\mathbb{F}_{2^n}$, we can naturally identify each element with an n-bit string.

▶ **Definition 5** (character). *A map $\chi : \mathbb{F}_{2^n} \to \mathbb{F}_2$ is called an* additive character *if for all $x, y \in \mathbb{F}_{2^n}$, $\chi(x + y) = \chi(x) + \chi(y)$. It is nontrivial if it is not the zero function.*

Since $\mathbb{F}_{2^n}$ is an n-dimensional vector space, we see the valuations on n basis vectors uniquely define the character. Consequently there are 2^n such characters. Notice we can conveniently characterize all characters either by $\chi_c(x) = \langle x, c \rangle$, or by fixing some character χ, and then defining $\chi_c(x) := \chi(c \cdot x)$. This can be seen by verifying these maps are characters, are distinct, and that there are 2^n of them (the latter is obvious since there are 2^n values of c).

3.3 Models of Computation

▶ **Definition 6** ($\mathbb{F}_2$-polynomials). *An $\mathbb{F}_2$-polynomial (or polynomial for short) is a function of the form $p(x) := \sum_{S \subset [n]} c_S \prod_{i \in S} x_i$ for some $c_i \in \mathbb{F}_2$ (all arithmetic here are over $\mathbb{F}_2$).*

▶ **Definition 7** (set-multilinearity). *An $\mathbb{F}_2$-polynomial p is* set-multilinear *over a partition $(X_1, \ldots, X_d)$ of variables if every monomial of p contains at most one variable from each X_i. Notice that all polynomials are trivially set-multililinear over $(x_1, \ldots, x_n)$.*

▶ **Definition 8** (junta). *Define the class $\mathsf{JUNTA}_{n,k}$ to be a function $\phi : \{0,1\}^n \to \{0,1\}$ which is solely dependent on k input bits (i.e. can be written as $\phi'(x_i)_{i \in S}$ for some subset $S \subset [n]$ of size k). Define $\mathsf{JUNTA}_{n,k}^{\oplus t}$ to be the class of functions which is the parity of t k-juntas.*

▶ **Definition 9** (k-party NOF protocol). *A boolean function $f : (\{0,1\}^{n/d})^d$ can be computed by a k-party NOF protocol with c bits of communication if on input $X = (X_1, \ldots, X_d)$, d players, can take turns writing a bit on the board, where player i's bit can only depend on X_{-d} and the other bits on the board, and the cth bit written is $f(X)$. We denote this class of functions to be Π_k^c.*

Circuits

We measure the size of a circuit by the total number of wires (including input wires) in it. AC^0_d are depth d circuits with unbounded fan-in whose gate set is $\{\mathsf{AND}, \mathsf{OR}, \mathsf{NOT}\}$. SYM is a gate which computes an arbitrary symmetric function, and THR is a gate which computes an arbitrary linear threshold function. In general, if we have a gate G, a subscript G_k will refer to its fan-in (in this case, G is fixed to have fan-in k).

▶ **Definition 10** ($(d, \mathcal{C})$-tree). *Let d be an integer and $\mathcal{C}$ a computational model (e.g. a circuit class). A function is computable by a $(d, \mathcal{C})$-tree if it is computable by a depth t decision tree with $\mathcal{C}$ functions as its leaves. That is, there exists a depth d decision tree T such that for every path π in T, $F|_\pi \in \mathcal{C}$.*

3.4 Probability

We will denote U_m to be the uniform distribution over the finite set $\{0,1\}^m$. We will also denote $S \subset_p T$ to be a random subset of T where each $t \in T$ is added to S independently with probability p.

▶ **Definition 11** (k-wise uniform). *Consider a distribution D over $(\{0,1\}^{n/d})^d$. We say that D is k-wise uniform if for all subsets $S = \{i_1, \ldots, i_k\} \subset [d]$ and all strings $y_1, \ldots, y_k \in \{0,1\}^{n/d}$,*

$$\Pr_{X \sim D}[\forall j, X_{i_j} = y_j] = 2^{-kn/d}.$$

▶ **Definition 12** (ε-close in distribution). *Let D_1 and D_2 be distributions over $\{0,1\}^n$. We say $D_1 \approx_\varepsilon D_2$, or equivalently D_1 is ε-close to D_2, if for all $S \subset \{0,1\}^n$,*

$$|\Pr_{x \sim D_1}[x \in S] - \Pr_{x \sim D_2}[x \in S]| \leq \varepsilon.$$

3.5 Random Restrictions and Partial Assignments

A *partial assignment* or *restriction* is a string $\rho \in \{0,1,\star\}^n$. Intuitively, a $\star$ represents an index that is still "alive" and hasn't been fixed to a value yet.

We also define a composition operation on partial assignments. For two restrictions ρ^1, ρ^2, define $\rho^1 \circ \rho^2$ so that

$$(\rho^1 \circ \rho^2)_i = \begin{cases} \rho^1_i & \rho^1_i \neq \star \\ \rho^2_i & \rho^1_i = \star. \end{cases}$$

Intuitively, one can see this as fixing bits determined by ρ^1 first, and then out of the remaining alive positions, fix them according to ρ^2.

A *random restriction* is simply a distribution over restrictions. A common random restriction we will use is R_p, the distribution where each index will be assigned $\star$ with probability p, and $0, 1$ each with probability $\frac{1-p}{2}$.

The main reason for defining restrictions is to observe their action on functions. Given a restriction ρ and function $f : \{0,1\}^n \to \{0,1\}$, we define $f|_\rho : \{0,1\}^n \to \{0,1\}$ to be the restricted function mapping $f|_\rho(x) := f(\rho \circ x)$.

3.6 Pseudorandomness

Our work will involve working with pseudorandomness primitives, like pseudorandom generators (PRGs) and randomness extractors (or simply extractors).

▶ **Definition 13** (ε-PRG). *A polytime computable function $G : \{0,1\}^s \to \{0,1\}$ is an ε-PRG for a subset $\mathcal{F}$ of functions $\{0,1\}^n \to \{0,1\}$ if for all $f \in \mathcal{F}$,*

$$|\mathsf{E}_{x\sim U_n}[(-1)^{f(x)}] - \mathsf{E}_{s\sim U_s}[(-1)^{f(G(s))}]| \leq \varepsilon.$$

We also say that G ε-fools $\mathcal{F}$. The parameter s is the seed length. In this paper, we will use a PRG of [27] which ε-fools $\mathbb{F}_2$ polynomials with $\leq S$ terms with seed length $2^{O(\sqrt{\log S})}\log(1/eps)$.

▶ **Definition 14** (min-entropy). *Let D be a distribution over $\{0,1\}^n$, and define $supp(D) = \{y \in \{0,1\}^n : \Pr_{x\sim D}[x = y] > 0\}$. Define the min-entropy of D to be the quantity*

$$-\log\left(\max_{x\in\{0,1\}^n} \Pr_{y\sim D}[y = x]\right).$$

It is helpful to note that if for a particular k and all $y \in \{0,1\}^n$, all probabilities $\Pr_{x\sim D}[x = y] \leq 2^{-k}$, then we know D has min-entropy $\geq k$.

▶ **Definition 15** (Strong/Linear/Seeded Extractors). *A (k,ε)-seeded extractor is a function $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ such that for any D with min-entropy $\geq k$, we have for $\boldsymbol{X} \sim D$ and $\boldsymbol{W} \sim U_d$ the following*

$$\mathsf{Ext}(\boldsymbol{X}, \boldsymbol{W}) \approx_\varepsilon U_m.$$

Ext is a strong seeded extractor if we also have

$$\Pr_{w\sim U_d}[\mathsf{Ext}(\boldsymbol{X}, w) \approx_\varepsilon U_m] \geq 1 - \varepsilon$$

Ext is a linear seeded extractor if for every fixed W, $\mathsf{Ext}(\cdot, W)$ is linear over $\mathbb{F}_2$. The Leftover Hash Lemma [16] allows us to construct a strong seeded (k,ε) extractor with seed length $2n$, $\mathsf{Ext} : \{0,1\}^n \cdot \{0,1\}^{2n} \to \{0,1\}^{k-2\log(1/\varepsilon)}$.

▶ **Definition 16** (Oblivious Bit-Fixing Source Extractors). *An (n,k) oblivious bit-fixing source (or OBF) is a distribution D over $\{0,1\}^n$ created by fixing some $n-k$ of the bits, and then filling in the remaining k indices with uniform and independent bits. An (k,ε) oblivious bit-fixing source extractor (or OBF extractor) is a function $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ such that for every (n,k) OBF D, we have that for $\boldsymbol{X} \sim D$,*

$$\mathsf{Ext}(\boldsymbol{X}) \approx_\varepsilon U_m.$$

For any $k > \sqrt{n}$, Kamp and Zuckerman [17] allows us to construct $(k, 2^{-\Omega(k^2/n)})$ OBF extractors $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^{\Omega(k^2/n)}$.

3.7 Correlation Bounds

We will need some tools and definitions from the literature of correlation bounds. We first give a formal definition of correlation.

▶ **Definition 17** (correlation). *For two Boolean functions $f, g : \{0,1\}^n \to \{0,1\}$, and a distribution D over $\{0,1\}^n$, define the correlation of f and g over D to be*

$$\mathsf{corr}_D(f,g) = |\mathsf{E}_{x\sim D}(-1)^{f(x)+g(x)}|.$$

If no distribution is mentioned, we always assume $D = U_n$. Furthermore, for a subset of functions $\mathcal{C}$, we define

$$\mathsf{corr}_D(f,\mathcal{C}) = \max_{g\in\mathcal{C}} \mathsf{corr}_D(f,g).$$

Viola and Wigderson defined a convenient quantity R_k, which is very useful in bounding correlations against NOF protocols.

▶ **Definition 18** (k-party Norm). *For a function $f : (\{0,1\}^{n/k})^k \to \{0,1\}$, define the k-party norm of f to be*

$$R_k(f) := \mathsf{E}_{X_1^{(0)},\ldots,X_k^{(0)},X_1^{(1)},\ldots,X_k^{(1)} \sim U_{n/k}} e\left(\sum_{\delta \in \{0,1\}^k} f(X_1^{(\delta_1)},\ldots,X_k^{(\delta_k)})\right).$$

This norm is useful due to the following theorem.

▶ **Theorem 19** ([31]). *Let $f : \{0,1\}^n \to \{0,1\}$ be arbitrary, and let g be computable by a d-party NOF protocol exchanging c bits. Then*

$$R_d(f) \leq \mathsf{corr}(f,g) \leq 2^c R_d(f)^{1/2^d}.$$

We will also use the following theorem of Nisan and Wigderson, which allow us to translate correlation bounds into PRGs.This version is seen in the survey of Hatami and Hoza [14]

▶ **Theorem 20** ([24], [14, Theorem 4.2.2]). *Let $f : \{0,1\}^n \to \{0,1\}$. Suppose $h : \{0,1\}^r \to \{0,1\}$ is ε-hard for $f \circ \mathsf{JUNTA}_{r,k}$ with respect to the uniform distribution. Then there exists a PRG for f with seed length $s = O(n^{\frac{1}{k+1}} \cdot r^2/k)$ and error εn.*

4 Nearly Optimal Correlation Bounds against $\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0$

We strictly improve upon the result [26] by proving a stronger correlation bound against $\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0$ circuits. This immediately gives PRGs against this class with improved seed length via the "hardness vs. randomness" framework [24] All previous work [28, 20, 26] looked at the function introduced in [25] created by taking the generalized inner product of parities. We present a new function comprised of field multiplication of extractors in order to prove stronger correlation bounds. Let m, n be parameters, and define $k := n/d$. We now prove the following result:

▶ **Theorem 21.** *Let $\mathsf{Ext} : \{0,1\}^k \to \{0,1\}^{.2k^{.996}}$ be a $(k^{.998}.2^{-.4k^{.996}})$ OBF-source extractor (explicit ones exist due to [17]). Let $f : (\{0,1\}^{.2k^{.996}})^d \to \{0,1\}$ be any function such that $\mathsf{corr}(f, \Pi_d^d) \leq 2^{-\Omega(k^{.996}/2^d)}$. Define $f \circ \mathsf{Ext}^d : (\{0,1\}^k)^d \to \{0,1\}$ to be the function*

$$f \circ \mathsf{Ext}^d(X) := f(\mathsf{Ext}(X_1),\ldots,\mathsf{Ext}(X_d)).$$

Let g be any function implementable by a $n^{O(\log n)}$-size $\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0$ circuit, and let $m = .0005 \log n$. Then

$$\mathsf{corr}(f \circ \mathsf{Ext}^{m+1}, g) \leq 2^{-\Omega(n^{.995})}.$$

In particular, by instantiating this template, say, with Ext being the extractor of [17] and f being either GIP [2] or FFM [10], we get explicit $f \circ \mathsf{Ext}^{m+1}$. We also note by simple adjusting of constants, we can get any $2^{-\Omega(n^{1-\varepsilon})}$ for constant $\varepsilon > 0$. This gives an improvement of the correlation bound given in [26] of $2^{-\Omega(n^{.499})}$.

Proof. We follow the same approach as done in [26]. The uniform distribution can be expressed as applying a random restriction, and then filling in the remaining bits uniformly. For good random restrictions, we argue that g simplifies to a $\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AND}_m$ circuit.

We then argue that even after the random restriction, $f \circ \mathsf{Ext}^{m+1}$ maintains its structural integrity due to the extractor. We then finish the argument by using Hastad and Goldmann's connection between $\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AND}_m$ and NOF protocols, and the fact that f has small correlation with $(m+1)$-party protocols.

The proof for the simplification of g is the same as seen in [26] so we merely cite it here. The only change is the tuning of parameters. Here is the lemma restated for our use.

▶ **Lemma 22.** *Let $g \in \{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0_d$ with circuit size $s = n^{\tau \log n}$. Then for $p = \frac{1}{48}(48 \log s)^{-(d-1)}$*

$$\begin{aligned}\Pr_{\rho \leftarrow \mathcal{R}_p}[g|\rho \text{ is not computed by } (.001pk, \{\mathsf{SYM}_{s^2}, \mathsf{THR}_{s^2}\} \circ \mathsf{AND}_{\log s})\text{-tree}] \\ &\leq s \cdot 2^{-.001pk/2^d} \\ &= 2^{-\Omega_d(pk)}\end{aligned}$$

Notice that for constant d this gives a bound of $2^{-\Omega(n/\mathsf{polylog}(n))}$, versus its use in [26] in which a $2^{-\Omega(\sqrt{n/\log n})}$ error was gained. We will see later that we can liberally set parameters here because our hard function maintains integrity even after traversing down a path of size $n/\mathsf{polylog}(n)$ (equivalent to randomly fixing $n/\mathsf{polylog}(n)$ bits), whereas the previous GIP function could only withstand $\sqrt{n}$ bits. This is result of using an OBF extractor with much better parameters than simply taking the XOR of many copies.

The leaves of our tree is now much simpler class of circuits, but it is not simple enough. Our correlation bounds can only handle circuits with fan in $m = O(\log n)$, but we currently have fan in $\log s = O(\log^2 n)$. Fix a leaf ℓ of the tree, and let $\{C_1, \ldots, C_{s^2}\}$ be a collection of subsets of $[n]$ where C_i contains the $\leq \log s$ indices of the variables that feed into the ith $\mathsf{AND}_{\log s}$ gate in the bottom layer. We now use the following basic fact, as in [20] and [26], that there is a large subset of variables that minimally intersect with each C_i.

▷ Claim 23. A random $\mathbf{L} \subset_q [n]$ (add each element to $\mathbf{L}$ with probability q) satisfies

$$\Pr[\exists i \in [s^2] \text{ such that } |C_i \cap \mathbf{L}| > m] \leq s^2 \binom{w}{m} q^m.$$

Instantiating this claim with our parameter setting of m and s, and setting $q = \Theta(n^{-.001})$ tells us

$$\Pr[\exists i \in [s^2] \text{ such that } |C_i \cap \mathbf{L}| > m] \leq \frac{1}{s}.$$

Hence there exists such an $L = L(\ell)$ such that restricting all bits outside L makes only $\leq m$ variables feed into each AND gate as desired.

To summarize, our restriction ρ is sampled by a distribution D specified by these three steps.

1. We first perform restriction $\mathcal{R}_p$,
2. and then randomly restrict $\leq .001pk$ while walking down the depth-$.001pk$ tree to a leaf ℓ,
3. and then randomly restrict all the variables alive in this leaf that is *not* in the $L(\ell)$ set that we showed existed

At the end of this process, we have by the union bound that with all but $2^{-\Omega(-pk)}$ probability, $g|_\rho$ becomes a $\{\mathsf{SYM}_{s^2}, \mathsf{THR}_{s^2}\} \circ \mathsf{AND}_m$ circuit.

We now observe what happens to $f \circ \mathsf{Ext}^{m+1}$ under this restriction ρ. We claim $f \circ \mathsf{Ext}^{m+1}$ retains its structure. Our wish is for at least $k^{.998}$ bits in each block to survive. That way, we will have a high entropy oblivious bit-fixing source fed into each extractor, and the function will be able to continue to strongly uncorrelate with m-party protocols. In Step 1, we draw a restriction from $\mathcal{R}_p$. Notice the live variables are distributed like a set $S \subset_p [n]$. We see that by a simple Chernoff and union bound,

$$\Pr_{\mathbf{S} \leftarrow \mathcal{R}_p} \left[\exists i \in [m+1] \text{ such that } |X_i \cap \mathbf{S}| < \frac{pk}{2} \right] \leq (m+1)2^{-\Omega(pk)}$$

Hence except for probability $m2^{-\Omega(pk)} = 2^{-\Omega(n^{1-o(1)})}$, each block X_i will have $\geq pk/2$ live variables. Conditioned on this, when we follow Step 2 and perform a random walk down the decision tree to a leaf, we will assign at most $.001pk$ bits, so we are guaranteed that each block X_i will contain at least $.499pk$ live variables. Step 3 is to take set $L(\ell)$ and arbitrarily restrict variables outside of it. We showed there exists an $L(\ell)$ which minimally overlaps with the input variables to the $\mathsf{AND}_{\log s}$ gates, but we want it to simultaneously overlap heavily with each block. That way most of the X_i will stay alive after restricting the bits outside of $L(\ell)$ The existence of such an $L(\ell)$ can be established by "completing the probabilistic method" started a few paragraphs above. Conditioning on good restrictions so far, let Y_i denote the variables that survived in X_i (hence $|Y_i| \geq .499pk$). We see that

$$\Pr_{\mathbf{L} \subset_q [n]} \left[\exists i \in [m+1] \text{ such that } |Y_i \cap \mathbf{L}| < \frac{.499pqk}{2} \right] \leq (m+1)2^{-\Omega(pqk)}.$$

Hence, the probability that $\mathbf{L}$ either intersects some C_i too much or some Y_i too little will happen with probability $\leq \frac{1}{s} + (m+1)2^{-\Omega(pqk)} \ll 1$. Thus there exists an $L(\ell)$ such that restricting all variables outside of it will simultaneously simplify g to a $\{\mathsf{SYM}_{s^2}, \mathsf{THR}_{s^2}\} \circ \mathsf{AND}_m$ and also leave at least $\frac{.499pqk}{2} \geq .249k^{.999}/\mathsf{polylog}(n) \gg k^{.998}$ variables alive. Stringing all three steps together, we know that except with probability $2^{-\Omega(-pk)}$, our random restriction ρ reduces g to $\{\mathsf{SYM}_{s^2}, \mathsf{THR}_{s^2}\} \circ \mathsf{AND}_m$, while simultaneously keeping $\geq k^{.998}$ variables in each X_i block alive.

We are now in the final phase of the argument where we now directly bound the correlation against the simplified circuit. We first state the results that will convert our circuits to NOF protocols.

▶ **Theorem 24** ([13]). *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function computed by a size-s $\mathsf{SYM} \circ \mathsf{AND}_m$ circuit. Then for any partition of the n inputs of f into $m+1$ blocks, there is a deterministic NOF $(m+1)$-party communication protocol that computes f using $O(m \log s)$ bits of communication.*

▶ **Theorem 25** ([23]). *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function computed by a $\mathsf{THR} \circ \mathsf{AND}_m$ circuit. Then for any partition of the n inputs of f into $m+1$ blocks, there is a randomized NOF $(m+1)$-party communication protocol that computes f with error γ_{err} using $O(m^3 \log n \log(n/\gamma_{err}))$ bits of communication.*

We now need to show an average-case hardness result for $f \circ \mathsf{Ext}^{m+1}|_\rho$ against NOF protocols. To do so, we will first calculate the k-party norm of $f \circ \mathsf{Ext}^{m+1}|_\rho$.

▶ **Lemma 26.** *Let ρ be a restriction which keeps $\geq k^{.998}$ variables in each X_i alive. Then $R_{m+1}(f \circ \mathsf{Ext}^{m+1}|\rho) \leq R_{m+1}(f) + 4(m+1) \cdot 2^{-4k^{.996}}$*

Proof. Now notice that

$$R_{m+1}(f \circ \mathsf{Ext}^{m+1}|_\rho) = \mathsf{E}_{X^{(0)},X^{(1)}} e\left(\sum_{\delta\in\{0,1\}^{m+1}} f(\mathsf{Ext}(X_1^{(\delta_1)}|_\rho),\ldots,\mathsf{Ext}(X_{m+1}^{(\delta_{m+1})}|_\rho))\right) \quad (2)$$

By assumption of ρ, each $X_i^{(\delta_i)}|\rho$ over uniform X_i is an OBF source with min-entropy $k^{.998}$, and so each $\mathsf{Ext}|_\rho(X_i) \approx_{2^{-4k^{.996}}} U_{.2k^{.996}}$. Since all $X_i^{(b)}$ for $i \in [m+1], b \in \{0,1\}$ are mutually independent, it follows by a hybrid argument that

$$(\mathsf{Ext}|_\rho(X_i^{(b)}|_\rho)_{i\in[m+1],b\in\{0,1\}} \approx_{2(m+1)2^{-4k^{.996}}} (U_{.2k^{.996}})_{i\in[m+1],b\in\{0,1\}}.$$

Therefore, we can upper bound Equation 2 by

$$\mathsf{E}_{(Y_i^{(b)})_{i\in[m],b\in\{0,1\}}} e\left(\sum_{\delta\in\{0,1\}^{m+1}} f(Y_1^{(\delta_1)}),\ldots,Y_{m+1}^{(\delta_{m+1})})\right) + 4(m+1)2^{-4k^{.996}}$$
$$\leq R_{m+1}(f) + 4(m+1)2^{-4k^{.996}}$$

as desired. ◀

With this, we can show that $f \circ \mathsf{Ext}^{m+1}|_\rho$ uncorrelates against randomized multiparty protocols.

▶ **Theorem 27.** *Let $g : \{0,1\}^n \to \{0,1\}$ be a Boolean function, and let ρ be a restriction such that $X_i|_\rho$ has $\geq k^{.998}$ live variables for each i, and $g|_\rho$ can be computed by an $(m+1)$-party NOF randomized protocol with with $\leq c$ bits and with error γ. Then*

$$\mathsf{corr}(f \circ \mathsf{Ext}^{m+1}|_\rho, g|_\rho) \leq 2\gamma + 2^{c-\Omega(k^{.996}/2^m)}.$$

This proof is deferred to the full version.

We now have all the ingredients to finish. Say ρ is good if ρ keeps $\geq k^{.998}$ variables alive in each block X_i *and* $g|_\rho$ is computable by $\{\mathsf{SYM},\mathsf{THR}\} \circ \mathsf{AND}_m$. We have shown for $\rho \sim D$, this doesn't happen only with probability $2^{-\Omega(pk)}$. If $g|_\rho$ has a SYM gate at the top, then Theorem 24 says the $\mathsf{SYM} \circ \mathsf{AND}_m$ circuit can be computed by a deterministic NOF protocol over $X_1,\ldots,X_{m+1}$ using $O(m \log s)$ bits. Plugging this in to Theorem 27 tells us

$$\mathsf{corr}(f \circ \mathsf{Ext}^{m+1}|_\rho, g|_\rho) \leq 2^{m\log s - \Omega(k^{.996}/2^m)} \leq 2^{-\Omega(n^{.995})}.$$

If the top gate is a THR, use Theorem 25 with $\gamma_{err} = 2^{-n^{.997}}$ to get that the circuit is a randomized NOF protocol over $X_1,\ldots,X_{m+1}$ using $O(m^3 \log n \log(n/\gamma_{err})) = O(n^{.995})$ bits. Plugging this into Theorem 27 gives us a correlation bound of

$$\mathsf{corr}(f \circ \mathsf{Ext}^{m+1}|_\rho, g|_\rho) \leq 2^{n^{.995} - \Omega(k^{.996}/2^m)} \leq 2^{-\Omega(n^{.996})}.$$

In either case we get the same bound, so we can bound

$$\begin{aligned}\mathsf{corr}(f \circ \mathsf{Ext}^{m+1}, g) &= |\mathsf{E}_{\rho\sim D}\mathsf{E}_X(-1)^{f\circ\mathsf{Ext}^{m+1}|_\rho(X)+g|_\rho(X)}| \\ &\leq 2^{-\Omega(pk)} + \mathsf{E}_{\rho\sim D}[|\mathsf{E}_X(-1)^{f\circ\mathsf{Ext}^{m+1}|_\rho(X)+g|_\rho(X)}||\rho \text{ is good}] \\ &\leq 2^{-\Omega(pk)} + 2^{-\Omega(n^{.995})} = 2^{-\Omega(n^{.995})}.\end{aligned}$$

The theorem is proved. ◀

▶ Remark 28. We note that the original RW function instantiated with different parameters can also get the same strengthened correlation bound. This requires a more nuanced analysis than present in [26], and does not extend to general functions of the form $f \circ \mathsf{Ext}^{m+1}$ as it relies on the specific structure of GIP and $\bigoplus$.

To recap the argument for a size s circuit, we first use the multi-switching lemma to reduce to a depth-2 circuit of fan-in $\log s$. We then restrict more variables so that the fan-in reduces to $\sqrt{\log s}$. We then apply correlation bounds for $\sqrt{\log s}$-party protocols to get an error of $\exp(-n/2^{\sqrt{\log s}})$. If one trusts that this error is the bottleneck in the argument, one can imagine running through the above argument again with $s = n^{\Theta(1)}$ to get a better error.

▶ **Corollary 29.** *Let $g(X)$ be a function implementable by a size $s = n^{O(1)}$-size $\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0$ circuit, and let $m = 2^{\sqrt{\log n}}$. Define $k := n/(m+1)$, and let $\mathsf{Ext} : \{0,1\}^k \to \{0,1\}^{k/2^{O(\sqrt{\log n})}}$ be a $(k/2^{O(\sqrt{\log n})}, 2^{-k/2^{O(\sqrt{\log n})}})$-extractor constructed from [17]. Then*

$$\mathsf{corr}(f \circ \mathsf{Ext}^{m+1}, g) \leq 2^{-(n/2^{O(\sqrt{\log s})})}.$$

This refinement will be useful for our correlation bounds against branching programs in the next section. As the proof is extremely similar to the above, we defer the sketch to the full version.

From Theorem 21, we derive the following two theorems as well.

▶ **Theorem 30.** *There exists an ε-PRG against size-S $\{\mathsf{SYM}, \mathsf{THR}\} \circ \mathsf{AC}^0$ circuits with seed length $s = 2^{O(\sqrt{\log S})} + (\log(1/\varepsilon))^{2.01}$.*

▶ **Theorem 31.** *There is an efficient ε-PRG which fools $\mathsf{AC}^0[\mathsf{SYM}, n^{.999}, S]$ with seed length $2^{O(\sqrt{\log S})} + (\log(1/\varepsilon))^{2.01}$ and an ε-PRG which fools $\mathsf{AC}^0[\mathsf{THR}, n^{.499}, S]$ with seed length $2^{O(\sqrt{\log S})} + (\log(1/\varepsilon))^{4.01}$.*

The proofs of these theorems follow by applying the Nisan-Wigderson hardness to randomness approach, as well as the decision tree bootstrapping idea of [20]. The details are deferred to the full version of the paper.

5 PRGs against $(d, \mathsf{poly}(n), n)$-2BPs

In this section, we use fortified hard functions to establish strong correlation bounds against the XOR of juntas, $\mathsf{JUNTA}_{n,d}^{\oplus \mathsf{poly}(n)}$. These are then pushed through the Nisan-Wigderson "hardness vs. randomness" framework to create PRGs which can fool $(d, \mathsf{poly}(n), n)$-2BPs. We first establish the correlation bounds, and then we show that this implies our desired PRG.

5.1 Correlation Bounds Against $\mathsf{JUNTA}_{n,d}^{\oplus \mathsf{poly}(n)}$

This subsection is devoted to proving the following result.

▶ **Theorem 32.** *Let $m = d \log n$, let h be the hard function in Corollary 29 instantiated on $k := n/m$ bits, and let $\oplus_m : \{0,1\}^m \to \{0,1\}$ be the parity function on m bits. We then have*

$$\mathsf{corr}(h \circ \oplus_m^k, \mathsf{JUNTA}_{n,d}^{\oplus n^c}) \leq \exp\left(-\frac{n}{d2^{O(\sqrt{\log n})}}\right)$$

Proof. Consider arbitrary $g \in \mathsf{JUNTA}_{n,d}^{\oplus n^c}$. We will show that there exists a subset $T \subset [n]$ of variables such that upon fixing all variables outside T, g simplifies to a sparse polynomial, while at least one input variable in each $\oplus_m$ stays alive. Write $f = \sum_{i=1}^{n^c} \phi_i$, where each ϕ_i is a d-junta. Let $S_i \subset [n]$ be the indices of the variables that ϕ_i depends on. Pick $T \subset_{1/d} [n]$. For a fixed i, we can bound

$$\Pr_T[|T \cap S_i| \geq \ell] \leq \sum_{\substack{S \subset S_i \\ |S|=\ell}} \Pr_T[S \subset T] = \binom{d}{\ell} \left(\frac{1}{d}\right)^\ell \leq \exp(-\Omega(\ell \log \ell)) \leq 0.1 n^{-c}.$$

for $\ell = \Theta(\log n)$. Union bounding over all i, it follows that

$$\Pr_{\rho \sim \mathcal{R}_{1/d}}[\exists i, |T \cap S_i| \geq \ell] < 0.1. \tag{3}$$

Let $X_1, \ldots, X_k$ be the input blocks of size m feeding into h. We can easily calculate

$$\Pr_T[\exists i, X_i \cap T = \emptyset] \leq k(1-1/d)^m \leq k \exp(-m/d) = 1/m = o(1). \tag{4}$$

Union bounding Equation (3) and Equation (4), it follows that there exists a subset $T \subset [n]$ that simultaneously intersects at most ℓ variables alive in each junta ϕ_i, and intersects at least one variable in each X_i. By pruning out elements, we can assume WLOG that there is exactly one variable in each X_i.

Since a function over b bits can be written as an $\mathbb{F}_2$-polynomial with up to 2^b terms, it follows for any restriction ρ with $\rho^{-1}(\star) = T$, $\phi_i|_\rho$ is a polynomial with $2^\ell = n^{\Theta(1)}$ terms. Therefore, $f|_\rho$ is a polynomial with $n^{\Theta(1)}$ terms as well, which can be written as a $n^{\Theta(1)}$-sized $\mathsf{PAR} \circ \mathsf{AND}$ circuit. Furthermore, we know that $h \circ \oplus_m^k|_\rho$ is equivalent to h up to negations of the inputs. As $\mathsf{SYM} \circ \mathsf{AC}^0$ is invariant under shifts of the input, we can appeal to Corollary 29 and observe

$$\begin{aligned}\mathsf{corr}(h \circ \oplus_m^k, g) &= |\mathsf{E}_X(-1)^{h \circ \oplus_m^k(X) + g(X)}| \\ &\leq \mathsf{E}_{X_{\overline{T}}} |\mathsf{E}_{X_T}(-1)^{h \circ \oplus_m^k(X_T, X_{\overline{T}}) + g(X_T, X_{\overline{T}})}| \leq \exp\left(-(n/d)/2^{O(\sqrt{\log n})}\right)\end{aligned}$$ ◀

5.2 Constructing and Analyzing the PRG

With this correlation bound in hand, we can construct good PRGs against the XOR of juntas using the Nisan-Wigderson framework.

▶ **Corollary 33.** *There is an ε-PRG for $\mathsf{JUNTA}_{n,d}^{\oplus n^{\Theta(1)}}$ with seed length $s = 2^{O(\sqrt{\log n})} d^2 \log^2(1/\varepsilon)$*

The proof is a straightforward application of the Nisan-Wigderson framework that we defer to the full version.

Fooling the parity of juntas actually allow us to fool arbitrary functions of juntas as long as the function has low Fourier L_1 norm.

▶ **Theorem 34.** *Let G be an ε-PRG for $\mathsf{JUNTA}_{n,d}^{\oplus m}$, and let $f : \{0,1\}^m \to \{0,1\}$. Then G is an $\varepsilon \cdot L_1(f)$-PRG for $f \circ \mathsf{JUNTA}_{n,d}$.*

We also defer this proof to the full version.

Finally, as an application, we show PRGs against (d, t, n)-2BPs, branching programs over n bits with width 2, length t, and reads d bits at a time. We will use the fact that width-2 branching programs which read one bit at a time have low Fourier L_1 norm (a proof can be found in [14]).

▶ **Lemma 35.** *If f is a $(1,t,n)$-2BP, then $L_1(f) \leq (t+1)/2$.*

We now use the fact that a (d,t,n)-2BP can be represented by a normal width-2 branching program acting on juntas to prove that the PRG from Corollary 33 fools (d,t,n)-2BPs.

▶ **Theorem 36.** *There exists an ε-PRG for (d,n^c,n)-2BPs with seed length $s = 2^{O(\sqrt{\log n})} \cdot d^2 \log^2(n/\varepsilon)$.*

Proof. Given a (d,n^c,n)-2BP B, we note that at each vertex $v \in [2n^c]$ of B, the transition function is some d-junta ϕ_v which will map the d bits read at that vertex to the next vertex to move to. Now consider the $(1,n^c,2n^c)$-2BP B' defined with the same vertex set as B, and define the transition function for $v \in [2n^c]$ in B' to read the vth bit of the input, and then map to the node in the next layer labeled by that bit. It is easy to see by construction that $B(x) = B'(\phi_1(x),\ldots,\phi_{2n^c}(x))$, which is a function in $B' \circ \mathsf{JUNTA}_{n,d}$. By Theorem 34, this can be ε-fooled by an $(\varepsilon/L_1(B'))$-PRG for $\mathsf{JUNTA}_{n,d}^{\oplus 2n^c}$. Using the L_1 bound from Lemma 35 and the construction from Corollary 33, we see that such a PRG has seed length $2^{O(\sqrt{\log n})} d^2 \log^2(1/\varepsilon)$. ◀

▶ Remark 37. There is an alternative PRG construction using the Ajtai-Wigderson framework [1] which gives optimal dependence on d, but exponentially worse dependence on ε. This is presented in the full version of the paper.

6 Correlation Bounds Against Set-Multilinear Polynomials

Our correlation bound for set-multilinear polynomials follows from an instantiation of the following theorem.

▶ **Theorem 38.** *Let $d \leq n$ be an integer. Let $\mathsf{Ext} : \{0,1\}^{n/d} \times \{0,1\}^{2n/d} \to \{0,1\}^{k-2\log(1/\varepsilon)}$ be a strong linear seeded (k,ε)-extractor with seed length $2n/d$ created from the Leftover Hash Lemma [16], and let χ some nontrivial additive character of $\mathbb{F}_{2^{n/d}}$. Define $\mathsf{ExtFFM}_d : \{0,1\}^{n+2n/d} \to \{0,1\}$ to be*

$$\mathsf{ExtFFM}_d(X,W) = \chi\left(\prod_{i=1}^{d} \mathsf{Ext}(X_i,W)\right).$$

Let $g : \{0,1\} \to \{0,1\}^n$ be a function, and let $S_1,\ldots,S_d \subset [n/d]$ be subsets of size $\geq k$ such that for any restriction ρ created by arbitrarily fixing all bits in W and outside S_i in X_i for each i, $g|_\rho$ always becomes set multilinear in $X_1,\ldots,X_d$. We then have

$$\mathsf{corr}(\mathsf{ExtFFM}_d, g) \leq d\varepsilon + (d-1)\left(\frac{1}{2^k\varepsilon^2} + \varepsilon\right).$$

Proof. For brevity, we let $f := \mathsf{ExtFFM}_d$ in this proof. We will first split the correlation expectation into first randomizing over all restrictions ρ of the bits in X outside of $S_1,\ldots,S_d$, then over the seed W, and then over the remaining live variables denoted by the S_i, which we denote $X_1|_\rho,\ldots,X_d|_\rho$. Now let W_ρ be the set of seeds w such that $\mathsf{Ext}(X_i|_\rho,w) \approx_\varepsilon U_k$ for all i. As Ext is strong-seeded, it follows by a union bound that W_ρ cover all but a $d\varepsilon$ fraction of seeds. Thus one can write

$$\begin{aligned}
\mathsf{corr}(f,g) &= |\mathsf{E}_X(-1)^{f'(X)+g(X)}| \\
&\leq \mathsf{E}_{W,\rho}\left|\mathsf{E}_X(-1)^{f|_\rho(X,W)+g|_\rho(X,W)}\right| \\
&\leq d\varepsilon + \mathsf{E}_\rho \mathsf{E}_{w\in W_\rho} |\mathsf{E}_X(-1)^{f|_\rho(X,w)+g|_\rho(X,w)}| \qquad (5)
\end{aligned}$$

Now fix a partial assignment ρ and seed $w \in W_\rho$. For brevity, let $f(\cdot) := f|_\rho(\cdot, w)$, and similarly for g'. By assumption, g' is set-multilinear over X We now apply a similar argument showing up in [3]. Let α be a map taking linear forms $\sum_{i\in[n/d]} c_i X_{d,i}$ in X_d to its vector of coefficients $(c_i) \in \mathbb{F}_2^{n/d}$. Note that by this definition, for any linear form $\ell(X_d)$, $\langle \ell(X_d), X_d\rangle = \ell(X_d)$. Letting $e(x) = (-1)^x$. We then see

$$\begin{aligned}\left|\mathsf{E}_X(-1)^{f'(X)+g'(X)}\right| &= \left|\mathsf{E}_X e\left(f(X_i) + \sum_{i\in[d-1]} g_i(X_{-i}) + g_d(X_d)\right)\right| \\ &\le \mathsf{E}_{X_{[d-1]}}\left|\mathsf{E}_{X_d} e\left(\langle \alpha(f(X_i) + \sum_{i\in[d-1]} g_i(X_{-i})), X_d\rangle + g_d(X_{-d})\right)\right| \\ &\le \Pr_{X_{[d-1]}}\left[\alpha(f'(X) + \sum_{i\in[d-1]} g_i(X_{-i})) = 0\right] \qquad (6)\end{aligned}$$

where we used the facts that f' is linear in X_d (as Ext here is a linear seeded extractor), $g_d(X_{-d})$ is independent of X_d, and linear forms are perfectly unbiased if their coefficient vector is nonzero. We now repeatedly use the simple inequality that for a linear map $h : \mathbb{F}_2^m \to \mathbb{F}_2^k$ and $a \in \mathbb{F}_2^k$, $\Pr_x[h(x) = a] \le \Pr_x[h(x) = 0]$ as follows.

$$\begin{aligned}&\Pr_{X_{[d-1]}}\left[\alpha(f'(X) + \sum_{i\in[d-1]} g_i(X_{-i})) = 0\right] \qquad (7)\\ &= \mathsf{E}_{X_{[d-2]}} \Pr_{X_{d-1}}\left[\alpha(f'(X) + \sum_{i=1}^{d-2} g_i(X_{-i}))) = \alpha(g_{d-1}(X_{-(d-1)}))\right] \\ &\le \Pr_{X_{[d-1]}}\left[\alpha\left(f'(X) + \sum_{i=1}^{d-2} g_i(X_{-i}))\right) = 0\right] \\ &\le \cdots \\ &\le \Pr_{X_{[d-1]}}[\alpha(f'(X)) = 0] \qquad (8)\end{aligned}$$

To analyze this probability, we state a lemma whose proof is deferred to the full version.

▶ **Lemma 39.** *For a linear form $\ell(X_d)$, $\alpha(\ell(X_d)) = 0$ if and only if $\ell(X_d) = 0$ for all X_d.*

Therefore, by Lemma 39,

$$\Pr_{X_{[d-1]}}[\alpha(f'(X)) = 0] = \Pr_{X_{[d-1]}}\left[\forall X_d, \chi\left(\prod_{i=1}^{d} \mathsf{Ext}(X_i|_\rho, w)\right) = 0\right].$$

Clearly if $\prod_{i=1}^{d-1} \mathsf{Ext}(X_i|_\rho, W) = 0$, f' becomes identically zero. When this doesn't happen, the function becomes of the form $\chi(c \cdot \mathsf{Ext}(X_d|_\rho, w))$ for some nonzero $c \in \mathbb{F}_{2^{n/d}}$. We now claim that there must exist some $X_d|_\rho$ such that $\chi(c \cdot \mathsf{Ext}(X_d|_\rho, w))$. Notice that for exactly $2^{n/d-1}$ values of Y, $\chi(cY) = 0$. As $w \in W_\rho$, the probability that a random $X_d|_\rho$ has $\mathsf{Ext}(X_d|_\rho, w)$ hit one of these values must be $\ge 1/2 - \varepsilon > 0$, proving the claim. Therefore, in order for $\alpha(f'(X)) = 0$, it is necessary that $\prod_{i=1}^{d-1} \mathsf{Ext}(X_i|_\rho, W) = 0$. Therefore,

$$\begin{aligned}\Pr_{X_{[d-1]}}[\alpha(f'(X)) = 0] &\leq \Pr_{X_{[d-1]}}\left[\prod_{i=1}^{d-1} \mathsf{Ext}(X_i|_\rho, w) = 0\right] \\ &\leq \sum_{i=1}^{d-1} \Pr_{X_i}[\mathsf{Ext}(X_i|_\rho, w) = 0] \\ &\leq (d-1)\left(\frac{1}{2^{k-2\log(1/\varepsilon)}} + \varepsilon\right)\end{aligned}$$

Stringing the above with inequalities (5), (6), and (8), we find

$$\mathsf{corr}(\mathsf{ExtFFM}_d, g) \leq d\varepsilon + (d-1)\left(\frac{1}{2^k\varepsilon^2} + \varepsilon\right). \qquad \blacktriangleleft$$

As a very nice application of this structural theorem, we show that we can achieve exponentially small correlation against $n^{O(1)}$-degree polynomials which are set-multilinear over some partition of the input into up to $n^{1-O(1)}$ parts.

▶ **Corollary 40.** *Let g be a degree $< d$ polynomial which is set-multilinear over an arbitrary partition $(A_1, \ldots, A_c)$ of X into c parts. Then*

$$\mathsf{corr}(\mathsf{ExtFFM}_d, g) \leq 2^{-\Omega(n/cd)}.$$

Proof. For each $i \in [n/d]$, define S_i to be the largest set among $\{X_i \cap A_1, \ldots, X_i \cap A_c\}$ (arbitrarily pick one if there are ties). Notice that the sets $\{X_i \cap A_j\}_{j\in[c]}$ partition X_i, and $|X_i| = n/d$. Therefore, we know that each $|S_i| \geq \frac{n/d}{c} = \frac{n}{cd}$. We now claim that any restriction ρ formed by arbitrarily fixing all the bits in X_i which are outside S_i, for each i, will make $g|_\rho$ set-multilinear over $(X_1, \ldots, X_d)$. Assume for the sake of contradiction there existed some monomial in $g|_\rho(X)$ that contained 2 variables from some X_i. Since $S_i \subset X_i$ and $S_j \cap X_i = \emptyset$ for $j \neq i$, both of these variables had to have come from S_i. But note that $S_i = X_i \cap A_\ell \subset A_\ell$ for some ℓ, and we know no monomial has 2 terms from the same A_i by our assumption of g. This yields our desired contradiction.

Therefore, we can apply Theorem 38 on the sets (S_i) with $k = n/cd$ and $\varepsilon = 2^{-.1n/cd}$ to deduce that

$$\mathsf{corr}(f, g) \leq d2^{-.1n/cd} + (d-1)(2^{-.8n/cd} + 2^{-.1n/cd}) = 2^{-\Omega(n/cd)}. \qquad \blacktriangleleft$$

References

1 Miklos Ajtai and Avi Wigderson. Deterministic simulation of probabilistic constant depth circuits. In *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*, pages 11–19, 1985. doi:10.1109/SFCS.1985.19.

2 L. Babai, N. Nisan, and M. Szegedy. Multiparty protocols and logspace-hard pseudorandom sequences. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, pages 1–11, New York, NY, USA, 1989. Association for Computing Machinery. doi:10.1145/73007.73008.

3 Abhishek Bhrushundi, Prahladh Harsha, Pooya Hatami, Swastik Kopparty, and Mrinal Kumar. On Multilinear Forms: Bias, Correlation, and Tensor Rank. In Jarosław Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*, volume 176 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 29:1–29:23, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.APPROX/RANDOM.2020.29.

4 Jaroslaw Blasiok, Peter Ivanov, Yaonan Jin, Chin Ho Lee, Rocco A. Servedio, and Emanuele Viola. Fourier growth of structured $\mathcal{O}_2$-polynomials and applications. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPIcs*, pages 53:1–53:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.APPROX/RANDOM.2021.53.

5 Andrej Bogdanov, Zeev Dvir, Elad Verbin, and Amir Yehudayoff. Pseudorandomness for width-2 branching programs. *Theory of Computing*, 9(7):283–293, 2013. doi:10.4086/toc.2013.v009a007.

6 Eshan Chattopadhyay, Jesse Goodman, Vipul Goyal, Ashutosh Kumar, Xin Li, Raghu Meka, and David Zuckerman. Extractors and secret sharing against bounded collusion protocols. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1226–1242, 2020. doi:10.1109/FOCS46700.2020.00117.

7 Eshan Chattopadhyay and Jyun-Jie Liao. Hardness Against Linear Branching Programs and More. In Amnon Ta-Shma, editor, *38th Computational Complexity Conference (CCC 2023)*, volume 264 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:27, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2023.9.

8 Ruiwen Chen, Valentine Kabanets, Antonina Kolokolova, Ronen Shaltiel, and David Zuckerman. Mining circuit lower bound proofs for meta-algorithms. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 262–273, 2014. doi:10.1109/CCC.2014.34.

9 Gil Cohen and Igor Shinkar. The complexity of dnf of parities. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, ITCS '16, pages 47–58, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2840728.2840734.

10 Jeff Ford and Anna Gál. Hadamard tensors and lower bounds on multiparty communication complexity. *Comput. Complex.*, 22(3):595–622, 2013. doi:10.1007/s00037-012-0052-6.

11 Parikshit Gopalan, Raghu Meka, Omer Reingold, and David Zuckerman. Pseudorandom generators for combinatorial shapes. *SIAM Journal on Computing*, 42(3):1051–1076, 2013. doi:10.1137/110854990.

12 Svyatoslav Gryaznov, Pavel Pudlák, and Navid Talebanfard. Linear branching programs and directional affine extractors. In *Proceedings of the 37th Computational Complexity Conference*, CCC '22, Dagstuhl, DEU, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2022.4.

13 J. Hastad and M. Goldmann. On the power of small-depth threshold circuits. In *Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science*, pages 610–618 vol.2, 1990. doi:10.1109/FSCS.1990.89582.

14 Pooya Hatami and William Hoza. Theory of unconditional pseudorandom generators. *Electron. Colloquium Comput. Complex.*, TR23-019, 2023. URL: https://eccc.weizmann.ac.il/report/2023/019, arXiv:TR23-019.

15 Pooya Hatami, William M. Hoza, Avishay Tal, and Roei Tell. Fooling constant-depth threshold circuits (extended abstract). In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 104–115, 2022. doi:10.1109/FOCS52979.2021.00019.

16 R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, pages 12–24, New York, NY, USA, 1989. Association for Computing Machinery. doi:10.1145/73007.73009.

17 Jesse Kamp and David Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. *SIAM Journal on Computing*, 36(5):1231–1247, 2007. doi:10.1137/S0097539705446846.

18 Ilan Komargodski, Ran Raz, and Avishay Tal. Improved average-case lower bounds for demorgan formula size. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 588–597, 2013. doi:10.1109/FOCS.2013.69.

19 Xin Li and Yan Zhong. Explicit Directional Affine Extractors and Improved Hardness for Linear Branching Programs. In Rahul Santhanam, editor, *39th Computational Complexity Conference (CCC 2024)*, volume 300 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:14, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2024.10.

20 Shachar Lovett and Srikanth Srinivasan. Correlation bounds for poly-size ac0 circuits with n(1-o(1)) symmetric gates. In Leslie Ann Goldberg, Klaus Jansen, R. Ravi, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 640–651, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

21 M. Luby, B. Velickovic, and A. Wigderson. Deterministic approximate counting of depth-2 circuits. In *[1993] The 2nd Israel Symposium on Theory and Computing Systems*, pages 18–24, 1993. doi:10.1109/ISTCS.1993.253488.

22 Xin Lyu. Improved pseudorandom generators for ac0 circuits. In *Proceedings of the 37th Computational Complexity Conference*, CCC '22, Dagstuhl, DEU, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2022.34.

23 Noam Nisan. The communication complexity of threshold gates. Combinatorics, Paul Erdős is eighty, Vol. 1, 1993.

24 Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of computer and System Sciences*, 49(2):149–167, 1994. doi:10.1016/S0022-0000(05)80043-1.

25 Alexander Razborov and Avi Wigderson. w(log n) lower bounds on the size of depth-3 threshold cicuits with and gates at the bottom. *Information Processing Letters*, 45(6):303–307, 1993. doi:10.1016/0020-0190(93)90041-7.

26 Rocco A. Servedio and Li-Yang Tan. Luby-Velickovic-Wigderson Revisited: Improved Correlation Bounds and Pseudorandom Generators for Depth-Two Circuits. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018)*, volume 116 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 56:1–56:20, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.APPROX-RANDOM.2018.56.

27 Rocco A. Servedio and Li-Yang Tan. Improved pseudorandom generators from pseudorandom multi-switching lemmas. *Theory Comput.*, 18:1–46, 2022. URL: https://theoryofcomputing.org/articles/v018a004/, doi:10.4086/TOC.2022.V018A004.

28 Emanuele Viola. Pseudorandom bits for constant-depth circuits with few arbitrary symmetric gates. *SIAM Journal on Computing*, 36(5):1387–1403, 2007. doi:10.1137/050640941.

29 Emanuele Viola. The sum of d small-bias generators fools polynomials of degree d. In *2008 23rd Annual IEEE Conference on Computational Complexity*, pages 124–127, 2008. doi:10.1109/CCC.2008.16.

30 Emanuele Viola. Correlation bounds against polynomials. *Electron. Colloquium Comput. Complex.*, TR22-142, 2022. URL: https://eccc.weizmann.ac.il/report/2022/142, arXiv:TR22-142.

31 Emanuele Viola and Avi Wigderson. Norms, xor lemmas, and lower bounds for gf(2) polynomials and multiparty protocols. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*, pages 141–154, 2007. doi:10.1109/CCC.2007.15.

32 Thomas Watson. Pseudorandom generators for combinatorial checkerboards. In *2011 IEEE 26th Annual Conference on Computational Complexity*, pages 232–242, 2011. doi:10.1109/CCC.2011.12.

Approximate Unitary k-Designs from Shallow, Low-Communication Circuits



Nicholas LaRacuente
Department of Computer Science, University of Indiana Bloomington, IN, USA



Felix Leditzky
Department of Mathematics, University of Illinois Urbana-Champaign, IL, USA
Illinois Quantum Information Science and Technology Center, Urbana, IL, USA

Abstract

Random unitaries are useful in quantum information and related fields but hard to generate with limited resources. An approximate unitary k-design is a measure over an ensemble of unitaries such that the average is close to a Haar (uniformly) random ensemble up to the first k moments. A strong notion of approximation bounds the distance from Haar randomness in relative error: the weighted twirl induced by an approximate design can be written as a convex combination involving that of an exact design and vice versa. The main focus of our work is on efficient constructions of approximate designs, in particular whether relative-error designs in sublinear depth are possible. We give a positive answer to this question as part of our main results:

1. **Twirl-Swap-Twirl:** Let A and B be systems of the same size. Consider a protocol that locally applies k-design unitaries to A^k and B^k respectively, then exchanges ℓ qudits between each copy of A and B respectively, then again applies local k-design unitaries. This protocol yields an ϵ-approximate relative k-design when $\ell = O(k \log k + \log(1/\epsilon))$. In particular, this bound is *independent* of the size of A and B as long as it is sufficiently large compared to k and $1/\epsilon$.
2. **Twirl-Crosstwirl:** Let $A_1, \dots, A_P$ be subsystems of a multipartite system A. Consider the following protocol for k copies of A: (1) locally apply a k-design unitary to each A_p for $p = 1, \dots, P$; (2) apply a "crosstwirl" k-design unitary across a joint system combining ℓ qudits from each A_p. Assuming each A_p's dimension is sufficiently large compared to other parameters, one can choose ℓ to be of the form $2(Pk+1)\log_q k + \log_q P + \log_q(1/\epsilon) + O(1)$ to achieve an ϵ-approximate relative k-design. As an intermediate step, we show that this protocol achieves a k-tensor-product-expander, in which the approximation error is in $2 \to 2$ norm, using communication logarithmic in k.
3. **Recursive Crosstwirl:** Consider an m-qudit system with connectivity given by a lattice in spatial dimension D. For every $D = 1, 2, \dots$, we give a construction of an ϵ-approximate relative k-design using unitaries of spatially local circuit depth

   $$O\big((\log m + \log(1/\epsilon) + k \log k)k \operatorname{polylog}(k)\big) .$$

   Moreover, across the boundaries of spatially contiguous sub-regions, unitaries used in the design ensemble require only area law communication up to corrections logarithmic in m. Hence they generate only that much entanglement on any product state input.

These constructions use the alternating projection method to analyze overlapping Haar twirls, giving a bound on the convergence speed to the full twirl with respect to the 2-norm. Using von Neumann subalgebra indices to replace system dimension, the 2-norm distance converts to relative error without introducing system size. The **Recursive Crosstwirl** construction answers one variant of [1, Open Problem 1], showing that with a specific, layered architecture, random circuits produce relative error k-designs in sublinear depth. Moreover, it addresses [1, Open Problem 7], showing that structured circuits in spatial dimension D of depth $\ll m^{1/D}$ may achieve approximate k-designs.

2012 ACM Subject Classification Theory of computation → Quantum information theory

Keywords and phrases Approximate unitary designs, Quantum circuits, Representation theory, von Neumann algebras

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.69



© Nicholas LaRacuente and Felix Leditzky;
licensed under Creative Commons License CC-BY 4.0

16th Innovations in Theoretical Computer Science Conference (ITCS 2025).
Editor: Raghu Meka; Article No. 69; pp. 69:1–69:2



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Category Extended Abstract

Related Version *Full Version*: https://arxiv.org/abs/2407.07876 [2]

Funding *Nicholas LaRacuente*: NL acknowledges support via the IBM Postdoc program at the Chicago Quantum Exchange.
Felix Leditzky: FL acknowledges support by National Science Foundation grant no. 2137953.

Acknowledgements We acknowledge helpful email exchanges with Aram Harrow during the development of these results. Part of this work was completed during the workshop "Beyond i.i.d. in Information Theory 11" hosted by the University of Tübingen from July 31 to August 4, 2023. We thank Thomas Schuster, Jonas Haferkamp, and Hsin-Yuan Huang for sharing results from their concurrent and independent work on low-depth designs and pseudorandom unitaries [3].

References

1 Aram W. Harrow and Saeed Mehraban. Approximate Unitary t-Designs by Short Random Quantum Circuits Using Nearest-Neighbor and Long-Range Gates. *Communications in Mathematical Physics*, 401(2):1531–1626, 7 2023. doi:10.1007/s00220-023-04675-z.

2 Nicholas LaRacuente and Felix Leditzky. Approximate unitary k-designs from shallow, low-communication circuits. *arXiv preprint*, 2024. Full version of this manuscript. doi:10.48550/arXiv.2407.07876.

3 Thomas Schuster, Jonas Haferkamp, and Hsin-Yuan Huang. Random unitaries in extremely low depth. *arXiv preprint*, 2024. doi:10.48550/arXiv.2407.07754.

Facility Location on High-Dimensional Euclidean Spaces

Euiwoong Lee
University of Michigan, Ann Arbor, MI, USA

Kijun Shin
Seoul National University, South Korea

Abstract

Recent years have seen great progress in the approximability of fundamental clustering and facility location problems on high-dimensional Euclidean spaces, including k-Means and k-Median. While they admit strictly better approximation ratios than their general metric versions, their approximation ratios are still higher than the hardness ratios for general metrics, leaving the possibility that the ultimate optimal approximation ratios will be the same between Euclidean and general metrics. Moreover, such an improved algorithm for Euclidean spaces is not known for Uncapaciated Facility Location (UFL), another fundamental problem in the area.

In this paper, we prove that for any $\gamma \geq 1.6774$ there exists $\varepsilon > 0$ such that Euclidean UFL admits a $(\gamma, 1 + 2e^{-\gamma} - \varepsilon)$-bifactor approximation algorithm, improving the result of Byrka and Aardal [3]. Together with the $(\gamma, 1 + 2e^{-\gamma})$ NP-hardness in general metrics, it shows the first separation between general and Euclidean metrics for the aforementioned basic problems. We also present an $(\alpha_{\mathrm{Li}} - \varepsilon)$-(unifactor) approximation algorithm for UFL for some $\varepsilon > 0$ in Euclidean spaces, where $\alpha_{\mathrm{Li}} \approx 1.488$ is the best-known approximation ratio for UFL by Li [15].

2012 ACM Subject Classification Theory of computation → Facility location and clustering

Keywords and phrases Approximation Algorithms, Clustering, Facility Location

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.70

Related Version *Full Version*: `https://arxiv.org/abs/2501.18105` [14]

Funding *Euiwoong Lee*: Supported by organization NSF CCF-2236669 and a gift from Google.

1 Introduction

The (metric) Uncapacitated Facility Location (UFL) is one of the most fundamental problems in computer science and operations research. The input of the problem consists of a metric space (X, d), a set of facility locations $\mathcal{F} \subseteq X$, a set of clients $\mathcal{C} \subseteq X$, as well as facility opening costs $\{f_i\}_{i \in \mathcal{F}}$. The goal is open a subset of centers $S \subseteq \mathcal{F}$ to minimize the sum of the *opening cost* $(\sum_{i \in S} f_i)$ and the *connection cost* $(\sum_{j \in \mathcal{C}} \min_{i \in S} d(i, j))$. After intensive research efforts over the years [11, 12, 16, 15], the best approximation ratio is 1.488 [15] and the best hardness ratio is 1.463 [11].

As the objective function is the sum of two heterogeneous terms of the opening cost and the connection cost, the natural notion of *bifactor approximation* has been actively studied as well. Formally, given an instance of UFL, a solution $S \subseteq \mathcal{F}$ is called an (λ_f, λ_c)-approximation for some $\lambda_f, \lambda_c \geq 1$ if, for any solution $T \subseteq \mathcal{F}$, the total cost of S is at most $\lambda_f \cdot F^* + \lambda_c \cdot C^*$, where F^*, C^* denote the opening and connection cost of T respectively. In particular, the case $\lambda_f = 1$, also known as a λ_c-*Lagrangian Multiplier Preserving (LMP)* approximation, has been actively studied due to its connection to another fundamental clustering problem of k-Median. There is a $(2 - \varepsilon)$-LMP approximation for some $\varepsilon > 2 \cdot 10^{-7}$ [8], and any λ_c-LMP approximation for UFL can be translated to $1.307 \cdot \lambda_c$-approximation for k-Median [9].


© Euiwoong Lee and Kijun Shin;
licensed under Creative Commons License CC-BY 4.0



Generalizing the hardness of Guha and Khuller [11], Jain, Mahdian and Saberi [12] proved that no (λ_f, λ_c)-approximation polynomial-time algorithm exists for $\lambda_c < 1 + 2e^{-\lambda_f}$ unless $\mathbf{P} = \mathbf{NP}$. (Guha-Khuller's hardness ratio $\gamma \approx 1.463$ is exactly the solution of $\gamma = 1 + 2e^{-\gamma}$.) While the optimal value for λ_c is not known for small values of λ_f, Byrka and Aardal gave an algorithm that achieves an $(\lambda_f, 1 + 2e^{-\lambda_f})$-approximation for any $\lambda_f \geq 1.6774$ [3].

Euclidean spaces are arguably the most natural metric spaces for facility location and clustering problems. Formally, *Euclidean UFL* is a special case of UFL where the underlying metric is $(\mathbb{R}^k, \|.\|_2)$ for some dimension k. When $k = O(1)$, this problem admits a PTAS [5], while the problem remains APX-hard when k is part of the input [7].[1]

Recent years have seen active studies on related k-Means and k-Median on high-dimensional Euclidean spaces [1, 10, 4], so that the best-known approximation ratios for them are 5.912 and 2.406 respectively. While they are strictly lower than the best-known approximation ratios for general metric spaces (which are 9 and 2.613), they are still larger than the best-known hardness ratios for general metrics (which are $1 + 8/e \approx 3.943$ and $1 + 2/e \approx 1.73$) [12], which means that it is still plausible that the optimal approximation ratios for k-Median and k-Means are the same between Euclidean metrics and general metrics.

Our first result is the first strict separation between Euclidean and general metric spaces for UFL. In particular, we show that Euclidean UFL admits a $(1.6774, 1 + 2e^{-1.6774} - \varepsilon)$ approximation for some universal constant $\varepsilon > 0$, which is NP-hard to do in general metrics.

▶ **Theorem 1.** *There exists a* $(1.6774, 1+2e^{-1.6774}-\varepsilon)$*-approximation algorithm for Euclidean UFL for some* $\varepsilon \geq 3 \cdot 10^{-42}$.

By the result of Mahdian et al. [16], it implies an $(\gamma, 1 + 2e^{-\gamma} - \varepsilon e^{-(\gamma-1.6774)})$-approximation for any $\gamma \geq 1.6774$. Using this result, we are able to slightly improve the approximation ratio for the best-known ($\alpha_{\mathrm{Li}} \approx 1.488$)-unifactor approximation of Li [15].

▶ **Theorem 2.** *There exists a* $(\alpha_{\mathrm{Li}} - \varepsilon)$*-approximation algorithm for Euclidean UFL for some* $\varepsilon \geq 2 \cdot 10^{-45}$.

Recent years also have seen great progress on hardness of approximation for clustering problems in high-dimensional Euclidean spaces, including Euclidean k-Means and k-Median [6, 7]. We show that similar techniques extend to UFL as well, proving the APX-hardness.

▶ **Theorem 3.** *Euclidean UFL is APX-hard.*

2 High-level Plan

Our work is based on the framework of Byrka and Aardal [3] who achieved an optimal $(\lambda_f, 1 + 2e^{-\lambda_f})$-bifactor approximation for $\lambda_f \geq 1.6774$ in general metrics. We first review their framework. It is based on the following standard linear programming (LP) relaxation:

$$\begin{aligned}
\text{Minimize} \quad & \sum_{i \in \mathcal{F}, j \in \mathcal{C}} d(i,j) x_{ij} + \sum_{i \in \mathcal{F}} f_i y_i & \\
\text{subject to} \quad & \sum_{i \in \mathcal{F}} x_{ij} = 1 & \forall j \in \mathcal{C}, \\
& x_{ij} \leq y_i & \forall i \in \mathcal{F}, j \in \mathcal{C}, \\
& x_{ij}, y_i \in [0,1] & \forall i \in \mathcal{F}, j \in \mathcal{C}.
\end{aligned}$$

[1] While the cited paper only studies k-Median and k-Means, the soundness analysis in their Theorem 4.1 (of the arXiv version) can be directly extended to any number of open facilities k, implying APX-hardness of Euclidean UFL.

The dual formulation is as follows:

$$\begin{aligned}
\text{Maximize} \quad & \sum_{j \in \mathcal{C}} v_j & \\
\text{subject to} \quad & \sum_{j \in \mathcal{C}} w_{ij} \leq f_i & \forall i \in \mathcal{F}, \\
& v_j \leq w_{ij} & \forall i \in \mathcal{F}, j \in \mathcal{C}, \\
& w_{ij} \geq 0 & \forall i \in \mathcal{F}, j \in \mathcal{C}.
\end{aligned}$$

A feasible solution (x, y) induces the *support graph*, which is defined as the bipartite graph $G = ((\mathcal{F}, \mathcal{C}), E)$ where nodes $i \in \mathcal{F}$ and $j \in \mathcal{C}$ are adjacent iff the corresponding LP variable $x_{ij} > 0$. Two clients $j, j' \in \mathcal{C}$ are considered *neighbors* in G if they share the same facility.

Let (x^*, y^*) be a fixed optimal solution to the primal program. The overall cost is divided into the facility cost $F^* = \sum_{i \in \mathcal{F}} f_i y_i^*$ and the connection cost $C^* = \sum_{i \in \mathcal{F}, j \in \mathcal{C}} d(i,j) x_{ij}^*$. Our goal is to round this solution to obtain a solution S whose total cost is at most $\lambda_f F^* + \lambda_c C^*$; it is well known that it implies the (λ_f, λ_c)-approximation defined in the introduction by scaling [3], so let us redefine the (λ_f, λ_c)-approximation for the rest of the paper so that S is (λ_f, λ_c)-approximate if its total cost is $\lambda_f F^* + \lambda_c C^*$.

The opening cost and connection cost for individual clients can be further divided using the optimal LP dual solution (v^*, w^*). For each client $j \in \mathcal{C}$, the fractional connection cost is given by $C_j^* = \sum_{i \in \mathcal{F}} d(i,j) x_{ij}^*$, and the fractional facility cost is computed as $F_j^* = v_j^* - C_j^*$. The *irregularity* of the facilities surrounding $j \in \mathcal{C}$ is defined by

$$r_\gamma(j) = \frac{d(j, \mathcal{D}_j) - d(j, \mathcal{D}_j \cup \mathcal{C}_j)}{F_j^*}.$$

Similarly,

$$r'_\gamma(j) = (\gamma - 1) \cdot r_\gamma(j) = \frac{d(j, \mathcal{D}_j \cup \mathcal{C}_j) - d(j, \mathcal{C}_j)}{F_j^*}.$$

If $\sum_{i \in \mathcal{F}'} y_i^* = 0$, we set $d(j, \mathcal{F}') = 0$. Similarly, when $F_j^* = 0$, we define $r_\gamma(j) = 0$ and $r'_\gamma(j) = 0$. According to the definition, the following conditions hold: the irregularity $0 \leq r_\gamma(j) \leq 1$, the average distance to a close facility $C_j = d(j, \mathcal{C}_j) = C_j^* - r'_\gamma(j) \cdot F_j^*$, and the average distance to a distant facility $D_j = d(j, \mathcal{D}_j) = C_j^* + r_\gamma(j) \cdot F_j^*$. The maximum distance to a close facility is bounded by $M_j \leq D_j$.

Clustering of [3]. At a high level, clustering operates based on the support graph $G = ((\mathcal{F}, \mathcal{C}), E)$. For each $c \in \mathcal{C}$, let $N_c := \{c' \in \mathcal{C} : \exists f \in \mathcal{F} \text{ such that } (c, f), (c', f) \in E\}$ be the neighbor of c. The clustering algorithm iteratively selects some client c as a *cluster center*, put all its neighbors into the cluster, and proceed with the remaining clients. Eventually, all clients are partitioned into one of these clusters. After this, for each cluster, exactly one facility adjacent to cluster center is opened. This ensures that every client is connected to a facility that is not too far from them. Therefore, the criteria for choosing cluster centers and opening facilities will determine the quality of solution.

Starting with a fractional solution of the LP (x^*, y^*) and a parameter $\gamma \in (1, 2)$, [3] constructed the *facility-augmented solution* $(\bar{x}, \bar{y})$, where each y_i^* value is multiplied by γ and each client $j \in \mathcal{C}$ reconfigures its x_{ij}^* values to be fractionally connected to as close facilities as possible. (E.g., $\bar{x}_{ij} > 0$ implies $x_{ij}^* > 0$, but not vice versa.) With some postprocessing,

one can also assume that $\bar{x}_{ij} \in \{0, \bar{y}_i\}$ for every $i \in \mathcal{F}, j \in \mathcal{C}$. Then one can categorize every facility near $j \in \mathcal{C}$ into two types: *close* facilities $\mathcal{C}_j = \{i \in \mathcal{F} \mid \bar{x}_{ij} > 0\}$ and *distant* facilities $\mathcal{D}_j = \{i \in \mathcal{F} \mid \bar{x}_{ij} = 0 \text{ and } x^*_{ij} > 0\}$. This implies that as γ increases, the clusters become smaller, and more facilities are opened.

Let the *average distance* from $j \in \mathcal{C}$ to a set of facilities $\mathcal{F}' \subseteq \mathcal{F}$ be defined as $d(j, \mathcal{F}') = \big(\sum_{i \in \mathcal{F}'} d(i,j) y^*_i\big) / \big(\sum_{i \in \mathcal{F}'} y^*_i\big)$. Then Let $C_j := d(j, \mathcal{C}_j)$, $M_j := \max_{i \in \mathcal{C}_j} d(j,i)$, and $D_j := d(j, \mathcal{D}_j)$. We have $C_j \le M_j \le D_j$.

At this point, the support graph is defined by $(\bar{x}, \bar{y})$ solution. Intuitively, we choose the client j with the smallest $C_j + M_j$ as a new cluster center. Given this clustering, the standard randomized rounding procedure is as follows:

1. For each cluster center j, choose exactly one facility from its neighboring facility set $\{i \in \mathcal{F} : (i,j) \in E\}$ according to the $\bar{y}$ values. (Recall that the sum of these values is exactly 1.)
2. For any facility $i \in \mathcal{F}$ that is not adjacent to any cluster center in G, independently open i with probability $\bar{y}_i$.

Algorithm 1 GREEDY: [3]'s clustering algorithm.

Require: Support graph $G = ((\mathcal{F}, \mathcal{C}), E)$
  $L = \{\}$
  while $\mathcal{C} \neq \emptyset$ **do**
    $c \leftarrow \arg\min_{j \in \mathcal{C}} (C_j + M_j)$
    $L \leftarrow L \cup (c, N_c)$
    $\mathcal{C} \leftarrow \mathcal{C} \setminus N_c$
  end while
  return L

Let us consider one client $j \in \mathcal{C}$ and see how its expected connection cost can be bounded under the above randomized rounding. Byrka and Aardal [3] proved the following properties.

- The probability that at least one facility in $\mathcal{C}_j$ is opened is at least $1 - e^{-1}$.
- The probability that at least one facility in $\mathcal{C}_j \cup \mathcal{D}_j$ is opened is at least $1 - e^{-\gamma}$.
- Let client $j' \in \mathcal{C}$ be a neighbor of j in G. Then, either $\mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j) = \emptyset$ or the *rerouting cost* $d(j, \mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j)) \le d(j', j) + d(j', \mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j)) \le D_j + C_{j'} + M_{j'}$ holds. Especially, when j' is the cluster center of j, it is at most $C_j + M_j + D_j$. (Li [15] refined this bound to $C_j + (3-\gamma)M_j + (\gamma - 1)D_j$.)

Then, one can (at least informally) expect that the expected connection cost of j is at most $(1 - e^{-1})C_j + (e^{-1} - e^{-\gamma})D_j + e^{-\gamma}(D_j + C_j + M_j)$. It turns out that setting $\gamma \approx 1.6774$ (the solution of $e^{-1} + e^{-\gamma} - (\gamma - 1)(1 - e^{-1} + e^{-\gamma}) = 0$) ensures that this value is at most $(1 + 2e^{-\gamma})C^*_j$, proving their $(\gamma, 1 + 2e^{-\gamma})$-bifactor. (See Section 6 for the formal treatment of their analysis as well as our improvement.)

Exploit the Geometry of Euclidean Spaces. In order to strictly improve the approximation ratio, it is natural to attempt to find a cluster N and its center j' where the above inequality holds with some additional slack. Let $cost_{j'}(j) = d(j, \mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j))$. Intuitively, our goal is to find a cluster $N \subseteq \mathcal{C}$ with center j' such that

$$\sum_{j \in N} cost_{j'}(j) \le \sum_{j \in N} \big((1 - \varepsilon_1)C_j + (3 - \gamma)M_j + (\gamma - 1)D_j\big). \tag{1}$$

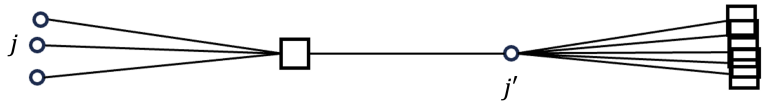


Figure 1 A simple case when $cost_{j'}(j) \approx 3$. There is a client-dense region on the left.

The only requirement from the rounding algorithm is that $N_{j'} \subseteq N$. Compared to [3]'s clustering, we want to shave $\varepsilon_1 \cdot C_j$ on average.

Let us consider the very special case where $C_j = M_j = D_j = 1$ for every $j \in \mathcal{C}$; every facility serving j in the original LP solution (x^*, y^*) is at the same distance from j. Let $j' \in \mathcal{C}$ be a cluster center and $j \in \mathcal{C}$ be in the cluster of j'. Then, a simple *3-hop triangle inequality* (just using $\mathcal{C}_j \cap \mathcal{C}_{j'} \neq \emptyset$) ensures that $cost_{j'}(j) \leq 3$, and our goal is to improve it to $(3 - \varepsilon_1)$. If $cost_{j'}(j) > 3 - \varepsilon_1$, how should the instance look like around j'?

It turns out that the instance around j' must exhibit a very specific structure in order to ensure that the 3-hop triangle inequality is tight for almost every neighbor $j \in N_{j'}$. We must have almost every $j \in N_{j'}$ located around almost the same point at distance 2 from j', where almost all facility neighbors of j' are at the opposite end of the line connecting $N_{j'}$ and j'. See Figure 1 for an example. Intuitively, the existence of such a *dense region* of clients suggests that if we let a client j'' in the region as a new center, many of the 3-hop-triangle inequalities cannot be tight, which implies average rerouting cost $cost_{j''}(j') \leq 3 - \varepsilon_1$. If j'' is again problematic, we can repeat this procedure over and over.

However, if we relax the condition to $C_j = M_j = 1 \leq D_j$, certain exceptions begin to emerge. One possible scenario is as follows: Since $cost_{j'}(j) = d(j, \mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j))$ captures the rerouting of j to j''s close facilities $\mathcal{C}_{j'}$ *except the j's facilities* $\mathcal{C}_j \cup \mathcal{D}_j$, if $\mathcal{C}_j \cup \mathcal{D}_j$ is large enough to exclude the facilities of $\mathcal{C}_{j'}$, then $cost_{j'}(j)$ might not behave as expected. However, a large volume of $\mathcal{C}_j \cup \mathcal{D}_j$ implies low C_j^*/F_j^* ratio. If the C^*/F^* ratio is sufficiently low, then this *facility-dominant* instance is actually easier to handle with a completely different algorithm, the JMS algorithm [12], which is known to be $(1.11, 1.7764)$-approximation algorithm.

Therefore, from now on, assume that the cluster centered at j' is connection-dominant. More strictly, assume that for any neighbor j of cluster center j' satisfies that $\mathcal{C}_j \cup \mathcal{D}_j$ cannot cover half of the ball centered at j' with a radius of 1. At this point, we can finally assert that it is impossible to avoid the formation of a dense region of clients.

Assume towards contradiction that there is no dense region and consider j in j''s cluster such that $cost_{j'}(j) > 3 - \varepsilon_1$. Almost all facilities of j' must be placed in one of two locations: on the opposite side of j relative to j', or within $\mathcal{C}_j \cup \mathcal{D}_j$. Since there is no dense region, there must be a neighbor k of j' such that $cost_{j'}(k) > 3 - \varepsilon_1$, and k is located in a different direction from j. This implies that the facilities positioned on the opposite side of j now help reduce $cost_{j'}(k)$, forcing that they are in $\mathcal{C}_k \cup \mathcal{D}_k$; since we assumed that $\mathcal{C}_k \cup \mathcal{D}_k$ cannot cover half of the unit ball around j', it implies the angle $\angle jj'k$ must be strictly greater than $\frac{\pi}{2}$! Ultimately, this process can be simplified to the following situation: inserting unit vectors into a unit sphere with every pairwise angle greater than $\frac{\pi}{2} + \varepsilon$ for some constant $\varepsilon > 0$. It is well known that in the geometry of Euclidean space, there is an upper bound $f(\varepsilon)$ on the number of such vectors, and such an upper bound shows that one of the regions around j (or k) we considered must have been dense. See Figure 2 for an example.

However, there are several technical barriers to extending this notion to the general case without restrictions on C_j, M_j, and D_j. In Section 3, we introduce these barriers and formalize the above concept. We also propose sufficient conditions to satisfy (1). From

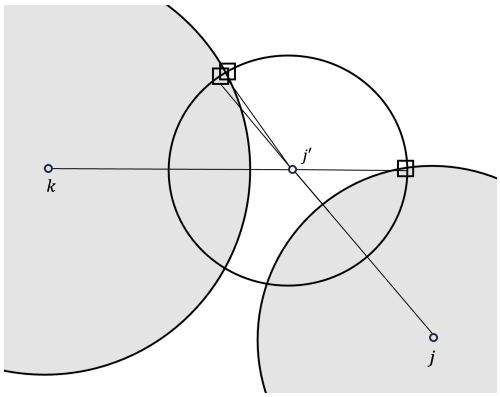


Figure 2 $\mathcal{C}_k \cup \mathcal{D}_k$ contains facilities positioned on the opposite side of j.

Section 3.1 to Section 5, we demonstrate how to remove these conditions, leaving only the connection-dominant instance assumptions. In Section 6, we propose and analyze the full algorithm, which achieves improved bi-factor approximation performance.

3 Finding Good Center via Geometry

In this section, we exploit the geometry of Euclidean spaces to prove the existence of a cluster center strictly better than the greedy choice of [3] under certain conditions (Theorem 10). We first define several concepts and introduce their motivation, including the sketch of our algorithm.

Recall that our goal is to find a cluster that satisfies (1). In all the following propositions, γ is a fixed value in the range $\gamma \in (1.6, 2)$.

▶ **Definition 4.** *Suppose $j' \in \mathcal{C}$ is a cluster center. Let $N_{j'}$ be the set of neighbors of j', and $\varepsilon_1 = 10^{-12}$. Additionally, define two more sets:*

$$N_{j'}^{-} = \{j \in N_{j'} \mid cost_{j'}(j) > (1-\varepsilon_1)C_j + (3-\gamma)M_j + (\gamma-1)D_j\}$$

$$N_{j'}^{+} = \{j \in \mathcal{C} \mid cost_{j'}(j) \leq (1-\varepsilon_1)C_j + (3-\gamma)M_j + (\gamma-1)D_j\}$$

*Moreover, **Saving** and **Spending** of center j' is defined as*

$$Saving(j') = \sum_{j \in N_{j'}^{+}} \{((1-\varepsilon_1)C_j + (3-\gamma)M_j + (\gamma-1)D_j) - cost_{j'}(j)\},$$

$$Spending(j') = \sum_{j \in N_{j'}^{-}} \{cost_{j'}(j) - ((1-\varepsilon_1)C_j + (3-\gamma)M_j + (\gamma-1)D_j)\}.$$

With the goal (1) in mind, $N_{j'}^{+}$ (resp. $N_{j'}^{-}$) contains clients j who meet (resp. do not meet) this goal, and $Saving(j')$ (resp. $Spending(j')$) indicates how much $cost_{j'}(j)$ exceeds (resp. is short of) this goal. Note that $N_{j'}^{-} \subseteq N_{j'} \subseteq N_{j'}^{+} \cup N_{j'}^{-}$ by definition; it is the *best* cluster for center j', which contains all its neighbors (which is required by the algorithm design) and possibly more to increase savings. Therefore, if $Saving(j') \geq Spending(j')$, then j' is considered a *good* center; otherwise, it is a *bad* center.

Now, we will explain some problematic situations that arise when extending the problem to the general case, i.e., without restrictions on C_j, M_j, and D_j's. The first simple concern is that the choice of the center j' will no longer be solely based on $C_{j'} + M_{j'}$ as in Algorithm 1,

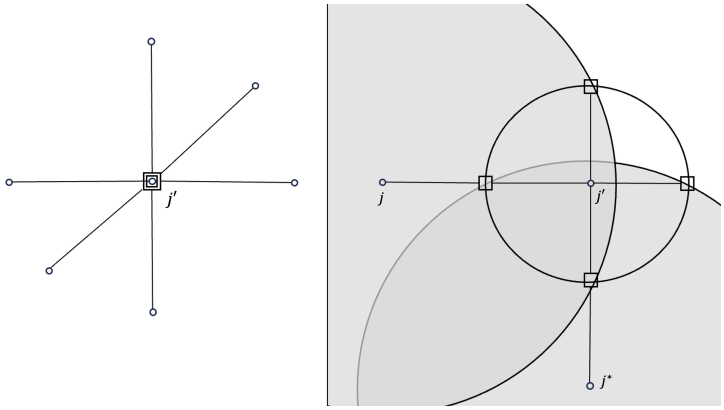


Figure 3 Two examples in a homogeneous network where every neighbor of center j' belongs to $N_{j'}^-$ without creating a dense region.

which breaks the previous arguments. Therefore, to gain more flexibility in selecting a new cluster center, it is beneficial to decompose the entire support graph into several layers, where each layer only concerns clients with roughly the same $C_{j'} + M_{j'}$ values.

▶ **Definition 5.** *A* ***network*** *is a subgraph* $((\mathcal{F}', \mathcal{C}'), E')$ *of the support graph* $G = ((\mathcal{F}, \mathcal{C}), E)$. *A network is called* ***homogeneous*** *if there exists* $s \geq 0$, *such that for any client* $j \in \mathcal{C}'$, $s \leq C_j + M_j \leq (1+\delta)s$ *for* $\delta = 3 \times 10^{-23}$.

However, there are still two more scenarios where the above strategy might not hold, as illustrated in Figure 3. This means that the neighbors in $N_{j'}^-$ are all bad clients, but do not create a dense region.

I. If $C_{j'} \ll s$, the facilities in $\mathcal{C}_{j'}$ are concentrated near j'. In this case, since all the facilities are very close to j', the 3-hop triangle inequality is almost tight for any $j \in N_{j'}$ regardless of where it is.
II. Recall that if $\mathcal{C}_j \cup \mathcal{D}_j$ is large enough to exclude the facilities of $\mathcal{C}_{j'}$, then $cost_{j'}(j)$ might not behave as expected. In particular, a technical problem arises when two facilities from $\mathcal{C}_{j'}$ that are almost antipodal with respect to j' are both contained in $\mathcal{C}_j \cup \mathcal{D}_j$, which is illustrated in Figure 3 (right).

The following definition addresses Scenario I.

▶ **Definition 6.** *For* $K_6 = 1.302$, *let* $\theta = \frac{K_6+1-\gamma}{2K_6+2-\gamma}$. *A client* $j \in \mathcal{C}$ *is* ***normal*** *if* $C_j \geq \theta(C_j + M_j)$, *otherwise* ***weird****.*

The following definition addresses Scenario II. Note that v_j^* is the maximum distance between j and any facility in $\mathcal{C}_j \cup \mathcal{D}_j$. We are interested in the ball around j' of radius $0.998 z_{j'}(j)$ (for sake of analysis), and j having a *small remote arm* with respect to j' means that $\mathcal{C}_j \cup \mathcal{D}_j$ cannot contain two antipodal points in our ball of interest (with some slack depending on α). Let $z_{j'}(j) = d(j', \mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j))$. The right-hand side represents the square of the length of the other side of a triangle, where the lengths of the two sides are $d(j', j)$ and $0.998 z_{j'}(j)$, and the included angle between them is $\frac{\pi}{2} - \alpha$.

▶ **Definition 7.** *For normal center* j', *a neighbor* j *of* j' *is said to have a* ***small remote arm*** *if the following holds for* $\alpha = 5 \times 10^{-4}$.

$$(v_j^*)^2 = (C_j^* + F_j^*)^2 < d(j', j)^2 + (0.998 z_{j'}(j))^2 - 2d(j', j) \cdot 0.998 z_{j'}(j) \cdot \cos\left(\frac{\pi}{2} - \alpha\right)$$

Otherwise, j *is said to have a* ***big remote arm****.*

In Section 3.1, we will show that these two scenarios are the only bad cases to worry about. For instance, if we assume that everything is normal and has a small remote arm, each candidate center $j' \in \mathcal{C}$ is either good or contains another candidate center j'' with $Saving(j'') > Saving(j')$ in its dense region.

How can we handle these two bad scenarios? In the following two lemmas, we prove that both the weird center and the big remote arm neighbor imply a high ratio of fractional facility cost $F^*_{j'}$ to fractional connection cost $C^*_{j'}$. The proof appears in the extended version of this paper [14].

▶ **Lemma 8.** *For any weird client j, $C^*_j \le K_6 F^*_j$ holds.*

▶ **Lemma 9.** *When $1.6 < \gamma < 2$, for a normal center j', if $j \in N^-_{j'}$ has a big remote arm, then $C^*_j \le K_6 F^*_j$.*

Therefore, if we consider a (sub)instance that has a low facility-to-connection cost ratio, it is natural to expect to apply this argument, which ultimately leads to a (somewhat) good center. Throughout the paper, we will often express this low facility-to-connection ratio condition will be expressed as $C^* > KF^*$ and use the following values for K: $K_1 = 1.3025 > K_2 = 1.3024 > K_3 = 1.3023 > K_4 = 1.3022 > K_5 = 1.3021 > K_6 = 1.302$. The following theorem is the main result of the section.

▶ **Theorem 10.** *Consider a homogeneous network and let j' be the normal center with the highest saving among all normal centers. Let $c^* = \sum_{j \in N^+_{j'} \cup N^-_{j'}} C^*_j$ and $f^* = \sum_{j \in N^+_{j'} \cup N^-_{j'}} F^*_j$. If $c^* > K_5 f^*$, then this cluster is **good on average** for $\varepsilon_2 = 5 \times 10^{-18}$ and every $\gamma \in (1.6, 2)$. i.e.*

$$\sum_{j \in N^+_{j'} \cup N^-_{j'}} cost_{j'}(j) \le \sum_{j \in N^+_{j'} \cup N^-_{j'}} ((1-\varepsilon_2)C_j + (3-\gamma)M_j + (\gamma-1)D_j).$$

3.1 Geometric Arguments

In this subsection, we prove Theorem 10 using properties of Euclidean geometry. As previously discussed, our goal is to show: When a bad cluster center j' is normal and (many of) its neighbors have a small remote arm, it is possible to find a dense region of clients near j'. When we consider the rerouting of j to j''s close facilities $\mathcal{C}_{j'}$, we can define some *worst* facilities. In the below definition, $B_{j'j} \subseteq \mathcal{C}_{j'}$ is the set of facilities that have the (almost) worst distance from j', and $T_{j'j} \subseteq B_{j'j}$ is the set of facilities with both worst distance and worst angle.

▶ **Definition 11.** *Let $r := 10^{-8}$, and let $\phi_r \approx 2 \times 10^{-4}$ be the minimum angle that satisfies $1 + x^2 + 2x\cos\phi_r \le (1 + (1-r)x)^2$ for $0 \le x \le 2$. For $j \in \mathcal{C}$ and its center j', let*

$$B_{j'j} := \{i \in \mathcal{C}_{j'} \mid 0.999 z_{j'}(j) \le d(i, j') \le M_{j'}\}$$
$$T_{j'j} := \{i \in B_{j'j} \mid \angle jj'i > \pi - \phi_r\}.$$

The following lemma shows that if j' is normal and $j \in N^-_{j'}$ has a bad rerouting through j', then among the facilities in $B_{j'j} \setminus (\mathcal{C}_j \cup \mathcal{D}_j)$, which are the rerouting candidates with worst distance, more than half of them must have a bad angle as well. It is a formalization of the intuition illustrated in Figure 1.

▶ **Lemma 12.** *For any $j \in N^-_{j'}$ of normal center j', $z_{j'}(j) > 0.99 C_{j'}$.*

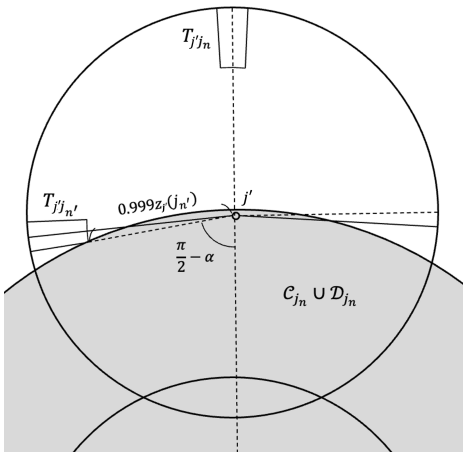


Figure 4 Intersection between a $T_{j'j}$ and a remote arm $\mathcal{C}_j \cup \mathcal{D}_j$.

Proof. By the triangle inequality and the homogeneous condition,

$$\begin{aligned} z_{j'}(j) \geq cost_{j'}(j) - d(j', j) &> (1-\varepsilon_1)C_j + (3-\gamma)M_j + (\gamma-1)D_j - (M_{j'} + M_j) \\ &\geq (1-\varepsilon_1)C_j + (2-\gamma)M_j + (\gamma-1)D_j - ((1+\delta)s - C_{j'}) \\ &\geq -(\varepsilon_1+\delta)s + C_{j'} \geq \left(1 - \frac{\varepsilon_1+\delta}{\theta}\right) C_{j'} \geq 0.99C_{j'}. \end{aligned}$$

◀

▶ **Lemma 13.** *For a normal center j' and any $j \in N_{j'}^-$, let $G_1 = T_{j'j} \cap (\mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j))$, $G_2 = (B_{j'j} \setminus T_{j'j}) \cap (\mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j))$. Then the following holds:*

$$\sum_{i \in G_1} y_i^* > \sum_{i \in G_2} y_i^*.$$

Proof. Assume the nontrivial case: $\mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j) \neq \emptyset$. Let $G_3 = (\mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j)) \setminus (G_1 \cup G_2)$. Let rerouting probability p_n and rerouting length l_n for $1 \leq n \leq 3$ as:

$$p_n = \frac{\sum_{i \in G_n} y_i^*}{\sum_{i \in \mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j)} y_i^*}, \quad l_n = \frac{\sum_{i \in G_n} y_i^* \cdot d(i, j')}{\sum_{i \in G_n} y_i^*}$$

Note that $p_1 + p_2 + p_3 = 1$ and $p_1 l_1 + p_2 l_2 + p_3 l_3 = z_{j'}(j)$. Then the goal of this theorem can be written as $p_1 > p_2$.

By Lemma 12, $z_{j'}(j) \geq 0.99C_{j'}$ holds. We derive a lower bound for $p_1 + p_2$ as:

$$z_{j'}(j) = p_1 l_1 + p_2 l_2 + p_3 l_3 \leq (p_1 + p_2) \cdot M_{j'} + (1 - p_1 - p_2) \cdot 0.999 z_{j'}(j)$$

implies that

$$p_1 + p_2 \geq \frac{0.001 z_{j'}(j)}{M_{j'} - 0.999 z_{j'}(j)} \geq \frac{0.001 \cdot 0.99 C_{j'}}{M_{j'}} \geq \frac{0.00099\theta}{1-\theta}$$

since the right-hand side gives the minimum value when $M_{j'}/C_{j'}$ is the maximum. It is bounded because j' is normal.

Denote a position vector as $\vec{v}$. From the definition of ϕ_r, $cost_{j'}(j)$ is at most

$$\begin{aligned} cost_{j'}(j) &= \frac{\sum_{i \in \mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j)} y_i^* \, ||(\vec{v_{j'}} - \vec{v_j}) + (\vec{v_i} - \vec{v_{j'}})||}{\sum_{i \in \mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j)} y_i^*} \\ &\leq p_1(d(j',j) + l_1) + p_2 \cdot (d(j',j) + (1-r)l_2) + p_3(d(j',j) + l_3) \\ &= d(j',j) + z_{j'}(j) - p_2 r l_2 \\ &\leq C_{j'} + M_{j'} + (2-\gamma)M_j + (\gamma - 1)D_j - p_2 r l_2 \\ &\leq (1 - 0.999 \cdot 0.99 p_2 r) C_{j'} + M_{j'} + (2-\gamma)M_j + (\gamma-1)D_j. \end{aligned}$$

Therefore, since $j \in N_{j'}^-$, $(1-\varepsilon_1)C_j + M_j < (1 - 0.999 \cdot 0.99 p_2 r)C_{j'} + M_{j'} \leq (1 - 0.98 p_2 r)C_{j'} + M_{j'}$ holds. It implies

$$\begin{aligned} \left(1 - \frac{\varepsilon_1}{2}\right)(C_j + M_j) \leq (1-\varepsilon_1)C_j + M_j &< (1 - 0.98 p_2 r)C_{j'} + M_{j'} \\ &\leq ((1 - 0.98 p_2 r)\theta + (1-\theta))(C_{j'} + M_{j'}). \end{aligned}$$

From the homogeneous condition,

$$\left(1 - \frac{\varepsilon_1}{2}\right) < (1 - 0.98 p_2 r \theta)(1 + \delta)$$

which implies

$$p_2 < \frac{1}{0.98\theta r} \cdot \frac{\delta + \frac{\varepsilon_1}{2}}{1+\delta} < \frac{0.00099\theta}{1-\theta} < p_1. \qquad \blacktriangleleft$$

The following argument demonstrates how the positional distribution of facilities in $\mathcal{C}_{j'}$ restricts that of the neighbors in $N_{j'}^-$. Consider two neighbors $j, k \in N_{j'}^-$, both with small remote arms, separated by an angle greater than $2\phi_r$, say $\frac{1}{100}$. Then, $T_{j'j} \cap T_{j'k} = \emptyset$. However, facilities in $T_{j'j}$ reduce $cost_{j'}(k)$ and vice versa, which implies that either $(\mathcal{C}_k \cup \mathcal{D}_k) \cap T_{j'j} \neq \emptyset$ or vice versa. As the small remote arm condition of j puts a limit on how much $(\mathcal{C}_j \cup \mathcal{D}_j)$ can intersect $\mathcal{C}_{j'}$, it is natural to expect that the number of such pairs is small.

▶ **Theorem 14.** *For a normal center j', let S be a subset of $N_{j'}^-$, consisting of clients with a small remote arm. Furthermore, let any two elements $j_1, j_2 \in S$ be separated by an angle greater than $\frac{1}{100}$ with respect to center j', i.e., $\angle j_1 j' j_2 > \frac{1}{100}$. Then, the cardinality of S is bounded by $M = 5 \times 10^6$, independent of the Euclidean space's dimension.*

Proof. Denote $S = \{j_1, j_2, ..., j_{|S|}\}$. Without loss of generality,

$$\sum_{i \in T_{j'j_k} \bigcap (\mathcal{C}_{j'} \setminus (\mathcal{C}_{j_k} \cup \mathcal{D}_{j_k}))} y_i^* \geq \sum_{i \in T_{j'j_{k+1}} \bigcap (\mathcal{C}_{j'} \setminus (\mathcal{C}_{j_{k+1}} \cup \mathcal{D}_{j_{k+1}}))} y_i^*$$

for all $1 \leq k < |S|$. Suppose $T_{j'j_{n'}} \cap (\mathcal{C}_{j_n} \cup \mathcal{D}_{j_n}) = \emptyset$ for some $n' < n$. Additionally, given $j_n \in N_{j'}^-$ and the small arm condition, it implies $T_{j'j_n} \cap (\mathcal{C}_{j_n} \cup \mathcal{D}_{j_n}) = \emptyset$. Therefore, $T_{j'j_{n'}} \cup T_{j'j_n} \subseteq \mathcal{C}_{j'} \setminus (\mathcal{C}_{j_n} \cup \mathcal{D}_{j_n})$. Moreover, $T_{j'j_{n'}} \cap T_{j'j_n} = \emptyset$ since $2\phi_r < \frac{1}{100}$. However, it contradicts to Lemma 13.

We show that two j's in S with similar $z_{j'}(j)$ values form a large angle with j'. Suppose $\angle j_{n'} j' j_n = \beta < \frac{\pi}{2} + \alpha - \phi_r$ for some $n' < n$ with $\frac{z_{j'}(j_n)}{z_{j'}(j_{n'})} \in [\frac{1}{1.001}, 1.001]$. Then for any point $x \in T_{j'j_{n'}}$, it holds that $d(j',x) \geq 0.999 z_{j'}(j_{n'}) > 0.998 z_{j'}(j_n)$ and $\angle j_n j' x < \beta + \phi_r < \frac{\pi}{2} + \alpha$. Also, the quadratic function $d(j',x)^2 - 2d(j',j_n)d(j',x) \cdot \sin(\alpha)$ is shown to be non-decreasing

for $d(j', x) > 0$. It comes from Lemma 12, which ensures that $d(j', x) > 0.998 z_{j'}(j_n) \geq 0.998 \cdot 0.99 C_{j'} \geq 0.98\theta s \geq 2(1+\delta)\sin\alpha \cdot s \geq d(j', j_n) \cdot \sin\alpha$. Therefore, since j_n has a small remote arm,

$$\begin{aligned} d(j_n, x)^2 &= d(j', j_n)^2 + d(j', x)^2 - 2d(j', j_n)d(j', x) \cdot \cos\angle j_n j' x \\ &> d(j', j_n)^2 + d(j', x)^2 - 2d(j', j_n)d(j', x) \cdot \sin\alpha \\ &\geq d(j', j_n)^2 + (0.998 z_{j'}(j_n))^2 - 2d(j', j_n) \cdot 0.998 z_{j'}(j_n) \cdot \sin\alpha \\ &> (C^*_{j_n} + F^*_{j_n})^2 = {v^*_{j_n}}^2 \end{aligned}$$

which implies $T_{j' j_{n'}} \bigcap (\mathcal{C}_{j_n} \cup \mathcal{D}_{j_n}) = \emptyset$, contradicting to the above result. Refer to Figure 4.

According to Rankin [17], the maximum number of disjoint spherical caps, each with an angular radius of $\frac{\pi}{4} + \frac{\alpha - \phi_r}{2}$, is at most $1 + \csc(\alpha - \phi_r)$ in any dimension. Since $0.99 C_{j'} \leq z_{j'}(j) \leq M_{j'}$ holds, it is feasible to segment this range into successive subranges such as $[0.99 C_{j'}, 1.001 \cdot 0.99 C_{j'}]$, $[1.001 \cdot 0.99 C_{j'}, 1.001^2 \cdot 0.99 C_{j'}]$, ..., up to $[M_{j'}/1.001, M_{j'}]$. Thus each subrange can only contain a finite number of clients with a small remote arm. Moreover, the normal center condition ensures a bounded number of such divisions. Consequently, the cardinality of $|S|$ is at most

$$|S| \leq \left(1 + \frac{1}{\sin(\alpha - \phi_r)}\right) \cdot \frac{\log \frac{M_{j'}}{0.99 C_{j'}}}{\log 1.001} \leq \left(1 + \frac{1}{\sin(\alpha - \phi_r)}\right) \cdot \frac{\log \frac{1-\theta}{0.99\theta}}{\log 1.001}. \quad \blacktriangleleft$$

Therefore, if we have a large $S \subseteq N^-_{j'}$ with small remote arms, there must exist a large subset of S whose pairwise angle is small, creating a dense region.

▶ **Lemma 15.** *For a normal center j', let S be a set of clients from $N^-_{j'}$ with a small remote arm. If $S \neq \emptyset$, then there exists a client $j \in S$ for which $Saving(j) \geq \frac{s}{125}\frac{|S|}{M}$.*

Proof. Let S' be a maximal subset of S where any two clients are separated by an angle greater than $\frac{1}{100}$ with respect to the center j'. Then, for any client $j \in S$, there exists a client $z_j \in S'$ such that $\angle j j' z_j \leq \frac{1}{100}$. Therefore, there exists a $z \in S'$ such that the number of clients $j \in S$ for which $z_j = z$ is at least $\frac{|S|}{|S'|}$.

For $1 \leq n \leq 5$, let R_n be a region where

$$R_n = \{x \in \mathbb{R}^l \mid \angle z\, j'\, x \leq \frac{1}{100},\ \frac{2n-2}{5}(1+\delta)s \leq d(j', x) \leq \frac{2n}{5}(1+\delta)s\}.$$

Therefore, there exists an index k such that $|R_k| \geq \frac{|S|}{5|S'|}$. For any two clients $j_1, j_2 \in R_k$, the distance $d(j_1, j_2)$ is bounded by the sum of their radial and angular differences. Hence, $d(j_1, j_2) \leq \frac{2(1+\delta)s}{5} + \frac{2(1+\delta)s}{50} = \frac{11}{25}(1+\delta)s$. From the triangle inequality and the homogeneous condition,

$$\begin{aligned} cost_{j_1}(j_2) &\leq d(j_1, j_2) + M_{j_1} \leq \frac{36}{25}(1+\delta)s \leq \frac{3-\varepsilon_1}{2(1+\delta)}s \\ &\leq (1-\varepsilon_1)C_{j_2} + 2M_{j_2} \leq (1-\varepsilon_1)C_{j_2} + (3-\gamma)M_{j_2} + (\gamma-1)D_{j_2} \end{aligned}$$

which implies $j_2 \in N^+_{j_1}$. By Theorem 14, $|S'| \leq M$. Therefore, the saving of any client $j \in R_k$ is at least

$$\begin{aligned} Saving(j) &\geq \frac{|S|}{5|S'|}\left((1-\varepsilon_1)C_j + (3-\gamma)M_j + (\gamma-1)D_j - \frac{36}{25}(1+\delta)s\right) \\ &\geq \frac{|S|}{5M}\frac{3 - 72\delta - 25\varepsilon_1}{50}s \geq \frac{s}{125}\frac{|S|}{M}. \end{aligned} \quad \blacktriangleleft$$

Given these geometric tools, Theorem 10 follows as the number of big-remote-arm-neighbors of the chosen center j' can be bounded.

▶ **Theorem 10.** *Consider a homogeneous network and let j' be the normal center with the highest saving among all normal centers. Let $c^* = \sum_{j \in N^+_{j'} \cup N^-_{j'}} C^*_j$ and $f^* = \sum_{j \in N^+_{j'} \cup N^-_{j'}} F^*_j$. If $c^* > K_5 f^*$, then this cluster is **good on average** for $\varepsilon_2 = 5 \times 10^{-18}$ and every $\gamma \in (1.6, 2)$. i.e.*

$$\sum_{j \in N^+_{j'} \cup N^-_{j'}} cost_{j'}(j) \leq \sum_{j \in N^+_{j'} \cup N^-_{j'}} ((1-\varepsilon_2)C_j + (3-\gamma)M_j + (\gamma-1)D_j).$$

The proof appears in the extended version of this paper [14].

4 Clustering for Homogeneous Instances

In this section, we present an algorithm that operates on a connection-dominant homogeneous instance, ensuring strictly better performance than a naive greedy clustering strategy. Let $c(i)$ for $i \in \mathcal{C}$ be the center of i when some clustering is given in the context.

▶ **Theorem 16.** *Suppose a homogeneous network $G = ((\mathcal{F}, \mathcal{C}), E)$ satisfies $C^* > K_4 F^*$. Then, the clustering produced by Algorithm 2 is **good on average**. Precisely, for $\varepsilon_3 = 3 \times 10^{-32}$ and every $\gamma \in (1.6, 2)$, the following holds:*

$$\sum_{j \in \mathcal{C}} cost_{c(j)}(j) \leq \sum_{j \in \mathcal{C}} ((1-\varepsilon_3)C_j + (3-\gamma)M_j + (\gamma-1)D_j).$$

Algorithm 2 HOMOGENEOUS: Homogeneous Clustering.

Require: Network $G = ((\mathcal{F}, \mathcal{C}), E)$
  $L = \{\}$
  while $\mathcal{C} \neq \emptyset$ **do**
    if normal client exists in $\mathcal{C}$ **then**
      $c \leftarrow \underset{j \text{ is normal}}{\arg\max}\ Saving(j)$
    else
      $c \leftarrow \underset{j \in \mathcal{C}}{\arg\min}\ (C_j + M_j)$
    end if
    $L \leftarrow L \cup (c, N^+_c \cup N^-_c)$
    $\mathcal{C} \leftarrow \mathcal{C} \setminus (N^+_c \cup N^-_c)$
  end while
  return L

For one of a cluster A made by the algorithm, if the center of A is weird, then the ratio is at most K_6 since all clients within them are weird. If A is not "good", its connection-facility ratio is at most K_5 by Theorem 10. Therefore, the assumed ratio K_4 in this theorem is larger than these values, implying that a constant proportion of clusters are "good" since they satisfy the conditions of Theorem 10. Therefore, by reducing the ε value by that proportion, the desired result can be obtained.

Proof. Divide $\mathcal{C}$ into three groups:

1. Clients clustered by a normal center, where the ratio of that cluster's connection cost to facility cost is greater than K_5.
2. Clients clustered by a normal center, where this ratio is at most K_5.
3. Clients clustered by a weird center.

Let S_n be the set of clients in the n-th group ($1 \le n \le 3$). Here, S_1 and S_2 correspond to clusters formed by the **if** condition, and S_3 is formed by the **else** condition. Define the following values:

$$C_n^* = \sum_{j \in S_n} C_j^*, \quad F_n^* = \sum_{j \in S_n} F_j^*$$

By Theorem 10, $\sum_{j\in S_1} cost_{c(j)}(j) \le \sum_{j\in S_1}((1-\varepsilon_2)C_j+(3-\gamma)M_j+(\gamma-1)D_j)$. The rerouting cost for S_2 is only bounded by the homogeneous condition. For a client j, $cost_{c(j)}(j) \le C_{c(j)}+M_{c(j)}+(2-\gamma)M_j+(\gamma-1)D_j \le (1+\delta)(C_j+M_j)+(2-\gamma)M_j+(\gamma-1)D_j$. Lastly, clients in S_3 are clustered through a greedy strategy. Thus, $cost_{c(j)}(j) \le C_j+(3-\gamma)M_j+(\gamma-1)D_j$. Note that S_3 consists solely of weird clients, meaning $C_3^* \le K_6F_3^* \le K_5F_3^*$.

From above, the total rerouting cost is bounded by:

$$\begin{aligned}
\sum_{j\in\mathcal{C}} cost_{c(j)}(j) &= \sum_{j\in S_1} cost_{c(j)}(j) + \sum_{j\in S_2} cost_{c(j)}(j) + \sum_{j\in S_3} cost_{c(j)}(j) \\
&\le \sum_{j\in S_1}((1-\varepsilon_2)C_j + M_j + (2-\gamma)M_j + (\gamma-1)D_j) \\
&\quad + \sum_{j\in S_2}((1+\delta)(C_j+M_j) + (2-\gamma)M_j + (\gamma-1)D_j) \\
&\quad + \sum_{j\in S_3}(C_j + (3-\gamma)M_j + (\gamma-1)D_j).
\end{aligned}$$

Therefore, it is sufficient to show that

$$(\delta+\varepsilon_3)\sum_{j\in S_2}(C_j+M_j) + \varepsilon_3\sum_{j\in S_3} C_j \le (\varepsilon_2-\varepsilon_3)\sum_{j\in S_1} C_j.$$

Note that $C_2^* + C_3^* \le K_5(F_2^* + F_3^*) < \frac{K_5}{K_4}C^*$ holds, which means $C_1^* > \left(1-\frac{K_5}{K_4}\right)C^*$. Also $C_1^* > K_4F_1^*$, since $C_2^* + C_3^* \le K_5(F_2^* + F_3^*)$. Then the following holds:

$$\begin{aligned}
&(\delta+\varepsilon_3)\sum_{j\in S_2}(C_j+M_j) + \varepsilon_3\sum_{j\in S_3} C_j \\
&\le (\delta+\varepsilon_3)\sum_{j\in S_2\cup S_3}(C_j+M_j) \le (\delta+\varepsilon_3)\sum_{j\in\mathcal{C}}(C_j+D_j) \\
&\le (\delta+\varepsilon_3)(2C^* + (2-\gamma)F^*) \\
&\le \frac{(\delta+\varepsilon_3)(2-\gamma+2K_4)}{K_4}C^* \\
&\le (\varepsilon_2-\varepsilon_3)\frac{(K_5-\gamma+1)(K_4-K_5)}{K_4K_5}C^* \le (\varepsilon_2-\varepsilon_3)\frac{K_5-\gamma+1}{K_5}C_1^* \\
&\le (\varepsilon_2-\varepsilon_3)(C_1^* - (\gamma-1)F_1^*) \le (\varepsilon_2-\varepsilon_3)\sum_{j\in S_1} C_j.
\end{aligned}$$

◀

5 Clustering for Connection-dominant Instances

In this section, we introduce an algorithm that operates on connection-dominant instances without a homogeneous condition. This algorithm uses Algorithm 1 and Algorithm 2 as subroutines. The theorem stated below shows our main result.

▶ **Theorem 17.** *For any connection-dominant instance, i.e. $C^* > K_1 F^*$, there exists an algorithm that finds a clustering configuration whose rerouting cost is at most*

$$\sum_{j\in\mathcal{C}} cost_{c(j)}(j) \le \sum_{j\in\mathcal{C}}((1-\varepsilon_5)C_j + (3-\gamma)M_j + (\gamma-1)D_j)$$

for $\varepsilon_5 = 2\times 10^{-41}$ and every $\gamma \in (1.6, 2)$.

▶ **Definition 18.** *Let B_0 be the set of clients for which $C_j + M_j = 0$. Let $s = \min_{j\in\mathcal{C}\setminus B_0}(C_j + M_j)$. For $\delta' = 7\times 10^{-32}$, let a **Block** B_n be the set of clients such that*

$$B_n = \{j \in \mathcal{C} \mid (1+\delta')^{n-1}s \le C_j + M_j < (1+\delta')^n s\}$$

Note that if $(1+\delta')^m \le (1+\delta)$, then a network composed of at most m consecutive blocks is still homogeneous. However, applying Algorithm 2 directly to each block would be impossible when the block is a facility-dominant or neighbors of the center from some block may belong to a different block. Moreover, the following fact implies that the neighbor relationships between consecutive blocks are the main point.

▶ **Observation 19.** *If two neighbors $j, j' \in \mathcal{C}$ belonging to neither the same block nor consecutive blocks, satisfying $(1+\delta')(C_j + M_j) \le C_{j'} + M_{j'}$, then $j' \in N_j^+$ holds when $Saving(\cdot)$ criteria ε satisfies that $\varepsilon \le \frac{2\delta'}{1+\delta'}$.*

Proof.

$$\begin{aligned} cost_j(j') &\le C_j + M_j + (2-\gamma)M_{j'} + (\gamma-1)D_{j'} \le \frac{C_{j'}+M_{j'}}{1+\delta'} + (2-\gamma)M_{j'} + (\gamma-1)D_{j'} \\ &\le \left(1 - \frac{2\delta'}{1+\delta'}\right)\cdot C_{j'} + (3-\gamma)M_{j'} + (\gamma-1)D_{j'}. \end{aligned}$$

◀

A key idea is that with a sufficiently small δ', the support graph can be segmented almost arbitrarily while still preserving the homogeneous condition, enabling the identification of weak *connections* between consecutive blocks. Notably, even if two clients $j \in B_n$ and $j' \in B_{n+1}$ come from consecutive blocks, the $cost_j(j')$ is not worse than the greedy strategy.

▶ **Definition 20.** *An **interval** I is the set of consecutive blocks, up to a maximum $2L$ for $L = 2\times 10^8$. Precisely, $I = \{B_i, \ldots, B_{i+k}\}$ for $k < L$, where $\sum_{j=i}^{i+k} C^*(B_j) - C^*(B_i) \ge K_3 \sum_{j=i}^{i+k} F^*(B_j)$, except when $k = 0$. The **size** of an interval I is the number of blocks it contains, denoted as $|I|$. The **reward** of interval I is defined as $R(I) = \sum_{j=i}^{i+k} C^*(B_j) - C^*(B_i)$.*

Note that a block is a unique type of interval with a size of 1. Also $(1+\delta')^{2L} \le 1+\delta$ holds. An interval is the basic unit of clustering to which Algorithm 1 and Algorithm 2 are applied. Therefore, the reward of an interval represents the extent to which a guaranteed *Saving* can be obtained when the current interval can be clustered using Algorithm 2, regardless of how the preceding intervals have been clustered. From this perspective, the first block, which does not contribute to the reward, serves as a kind of "buffer".

Given the entire support graph $G = (\mathcal{F} \cup \mathcal{C}, E)$ and a subset $\mathcal{C}' \subseteq \mathcal{C}$, let $G_{\mathcal{C}'}$ be the subgraph (network) induced by $\mathcal{F} \cup \mathcal{C}'$. Consequently, when we call Algorithms 1 or 2 with $G_{\mathcal{C}'}$, the calculations for *Saving* and *Spending* are performed solely with respect to the implicitly defined client set $\mathcal{C}'$. We denote Algorithm 1 as GREEDY and Algorithm 2 as HOMOGENEOUS. For simplicity, when I denotes an interval, we interpret the expression $\sum_{j \in I}$ as $\sum_{B \in I} \sum_{j \in B}$, aggregating over all clients within the interval. The following theorem illustrates how reward is related to *Saving*. The proof appears in the extended version of this paper [14].

▶ **Lemma 21.** *Let J be a set of non-overlapping intervals. Then clustering produced by Algorithm 3 is* ***good on average****. Precisely, for $\varepsilon_4 = 2 \times 10^{-36}$,*

$$\sum_{j \in \mathcal{C}} cost_{c(j)}(j) \leq \sum_{j \in \mathcal{C}} \left(\left(1 - \frac{\sum_{I \in J} R(I)}{\sum_{j' \in \mathcal{C}} C_{j'}} \varepsilon_4 \right) C_j + (3 - \gamma) M_j + (\gamma - 1) D_j \right).$$

Algorithm 3 CONN: Connection-dominant Clustering.

Require: Set of disjoint intervals $J = \{I_1, I_2, \ldots, I_m\}$, arranged in the increasing order of $C_j + M_j$ values of their clients.
  $L = \{\}$
  for $k = 1$ to m **do**
    if $|I_k| \geq 2$ and $\sum_{j \in I_k} C_j^* > K_4 \sum_{j \in I_k} F_j^*$ **then**
      $L \leftarrow L \cup \text{HOMOGENOUS}(G_{I_k})$
    else
      $L \leftarrow L \cup \text{GREEDY}(G_{I_k})$
    end if
    $S := \bigcup_{j \in I_k} \left(N_j^+ \cup N_j^- \right)$
    for $l = k$ to m **do**
      $I_l \leftarrow I_l \setminus S$
    end for
  end for
  return L

In conclusion, it suffices to find a set of non-overlapping intervals J which have a high $R(J)$ value. Here, we will briefly touch on the idea. We will iterate through the blocks in reverse order $(B_N, \ldots, B_0)$, cutting out suitable ranges that satisfy the interval conditions. Fix a point r to be the right end of the interval, and expand the left end of the current range one block to the left until the range is suitable for processing. Suppose, at some point, the C/F value of the current range is less than K_2, which is strictly less ratio of the input instance, K_1. Then, this range can be considered a minor part and can be excluded as there is no need for it to become an interval. Therefore, we only consider cases where the current range has a C/F value of at least K_2.

However, in the situation where the current range's reward is small, meaning that the first block must avoid most of the C^*. In this case, if we keep expanding the range to the left, we will eventually reach the initial block B_0, satisfying the interval condition. There remains a subtle issue of the range's length exceeding $2L$ during the expansion process, but this implies that the C^* value on the left side of the range is always exponentially increasing compared to the right side. This can be resolved by appropriately reducing the right side of the range.

▶ **Lemma 22.** *For any support graph G that is connection-dominant, $C^* > K_1 F^*$, then the set of non-overlapping intervals J obtained by Algorithm 4 satisfies $\sum_{I \in J} R(I) \geq \frac{1}{10^5} C^*$.*

Algorithm 4 CUTINTERVAL: Find a set of non-overlapping intervals J with large rewards.

Require: Network $G = ((\mathcal{F}, \mathcal{C}), E)$.
  $J := \emptyset$
  $r := N$
  while $r > 0$ **do**
    $l := r$
    $c := 0, f := 0$
    while $l \geq 0$ **do**
      if $c \geq K_3(f + F^*(B_l))$ and $C^*(B_l) \leq \frac{K_2 - K_3}{K_2} c$ **then**
        $J \leftarrow J \cup \{[l, r]\}$
        $r \leftarrow l - 1$
        break
      end if
      $c \leftarrow c + C^*(B_l), f \leftarrow f + F^*(B_l)$
      if $c < K_2 f$ **then**
        $r \leftarrow l - 1$
        break
      end if
      if $r - l + 1 = 2L$ **then**
        $c \leftarrow c - \sum_{i=l+L}^{r} C^*(B_i), f \leftarrow f - \sum_{i=l+L}^{r} F^*(B_i)$
        $r \leftarrow r - L$
      end if
      $l \leftarrow l - 1$
    end while
  end while
  Include in J every block (as an interval of size 1) not already part of J
  return J

The proof appears in the extended version of this paper [14].

By directly applying Lemma 21 and Lemma 22, Theorem 17 can be proved.

6 Improved Bifactor Approximation

In this section, we present an improved bifactor approximation algorithm for UFL, proving Theorem 1. To deal with facility-dominant instances, we employ the JMS algorithm [12], which is known to be $(1.11, 1.7764)$-approximation algorithm.

Now, we will show that Algorithm 5 satisfies Theorem 1. Let $\gamma_0 \leq 1.6774$ be a solution of the below equation.

$$\frac{1}{e} + \frac{1}{e^\gamma} - (\gamma - 1)(1 - \frac{1}{e} + (1 - \varepsilon_5)\frac{1}{e^\gamma}) = 0.$$

To prove Theorem 1, we consider two cases. When $C^* \leq K_1 F^*$, the inequality $1.11F^* + 1.7764C^* \leq 1.6774F^* + (1 + 2e^{-1.6774} - \varepsilon_6)C^*$ holds. For the case $C^* > K_1 F^*$, we can employ the same proof structure as the statement proved in [3][2]. The proof appears in the extended version of this paper [14].

[2] Theorem 4.3 of [3]

Algorithm 5 Overall bi-factor clustering process.

Derive $(\bar{x}, \bar{y})$ from the optimal LP primal solution (x^*, y^*).
if $C^* \leq K_1 F^*$ **then**
  Execute the $(1.11, 1.7764)$-approximation algorithm.
else
  Proceed with CONN(CUTINTERVAL(G)) (Algorithm 3 and 4).
  for all cluster centers j **do**
    Open exactly one nearby facility with a probability of $\bar{x}_{ij}$.
  end for
  for all facilities i not close to any cluster center **do**
    Open each facility with a probability of $\bar{y}_i$ independently.
  end for
  Connect each client to the nearest open facility.
end if

7 Improved Unifactor Approximation

In this section, we propose the algorithm that guarantees better unifactor approximation suggested by Li [15], proving Theorem 2.

Framework of [15]. Li showed that a hard instance for a certain γ might not be a hard instance for another value of γ. This suggests that selecting a γ value at random could improve the expected performance of the algorithm.

They introduced a *characteristic function* $h : (0, 1] \to \mathbb{R}$ to represent the distribution of distances between a client and its neighboring facilities. For a client $j \in \mathcal{C}$, assume $i_1, i_2, \ldots, i_k$ are the facilities within $\mathcal{C}_j \cup \mathcal{D}_j$, ordered by increasing distance from j. Then, for $0 < p \leq 1$, $h_j(p)$ is defined as $d(j, i_t)$, where t is the smallest index satisfying $\sum_{l=1}^{t} y^*_{i_l} \geq p$. Also, they improved the bound of the rerouting cost as $d(j, \mathcal{C}_{j'} \setminus (\mathcal{C}_j \cup \mathcal{D}_j)) \leq (2-\gamma)M_j + (\gamma-1)D_j + C_{j'} + M_{j'}$. For fixed γ, the expected connection cost for j for the aforementioned algorithm of [3] is at most

$$\mathbb{E}[C_j] \leq \int_0^1 h_j(p) e^{-\gamma p} \gamma \, dp + e^{-\gamma} \left(\gamma \int_0^1 h_j(p) \, dp + (3-\gamma) h_j(\frac{1}{\gamma}) \right).$$

Therefore, it can be modeled as a 0-sum game to analyze the approximation ratio. The characteristic function for the whole instance is given by $h(p) = \sum_{j \in \mathcal{C}} h_j(p)$. Assuming h is normalized so that $\int_0^1 h(p) \, dp = 1$, the algorithm proceeds as follows: with probability κ, it employs the JMS algorithm [12]. Mahdian [16] proved that the JMS algorithm achieves a $(1.11, 1.7764)$-approximation. Otherwise, γ is sampled randomly from the distribution $\mu : (1, \infty] \to \mathbb{R}^*$, ensuring that $\kappa + \int_1^\infty \mu(\gamma) \, d\gamma = 1$. Thus, the value of the 0-sum game, i.e., the approximation ratio of the algorithm under a fixed strategy (κ, μ), is calculated as follows:

$$\nu(\kappa, \mu, h) = \max\left\{ \int_1^\infty \gamma \mu(\gamma) \, d\gamma + 1.11\kappa, \int_1^\infty \alpha(\gamma, h) \mu(\gamma) \, d\gamma + 1.7764\kappa \right\}$$

where

$$\alpha(\gamma, h) = \int_0^1 h(p) e^{-\gamma p} \gamma \, dp + e^{-\gamma} \left(\gamma + (3-\gamma) h\left(\frac{1}{\gamma}\right) \right).$$

Moreover, for a given probability density function μ for γ, it can be shown that the characteristic function for the hardest instance is a *threshold function*, which defined as

$$h_q(p) = \begin{cases} \frac{1}{1-q}, & \text{for } p > q \\ 0, & \text{for } p \le q \end{cases}$$

for some $0 \le q < 1$. This means that the final approximation ratio for some μ is given by $\max_{0 \le q < 1} \nu(\kappa, \mu, h_q)$. In [15], the suggested distribution for α_{Li} is $\mu(p) = \theta D(p - \gamma_1) + \frac{1-\kappa-\theta}{\gamma_2-\gamma_1}[\gamma_1 < p < \gamma_2]$, where D is Dirac delta function, $\gamma_1 = 1.479311$, $\gamma_2 = 2.016569$, $\theta = 0.503357$, $\kappa = 0.195583$.

Our Improvement. By using Algorithm 5, the expected connection cost for some γ and characteristic function h is given as follows:

$$\alpha'(\gamma, h) = \begin{cases} \int_0^1 h(p) e^{-\gamma p} \gamma \, dp + e^{-\gamma} \left(\gamma(1-\varepsilon_7) + (3-\gamma) h(\frac{1}{\gamma}) \right), & \text{if } 1.6 < \gamma < 2 \\ \int_0^1 h(p) e^{-\gamma p} \gamma \, dp + e^{-\gamma} \left(\gamma + (3-\gamma) h(\frac{1}{\gamma}) \right). & \text{otherwise} \end{cases}$$

Therefore, we define a new 0-sum game value as follows, assuming h is scaled up such that $\int_0^1 h(p)\, dp = 1$.

$$\nu'(\mu, \theta, h) = \max\left\{ \int_1^\infty \gamma \mu(\gamma)\, d\gamma + 1.11\kappa, \int_1^\infty \alpha'(\gamma, h)\mu(\gamma)\, d\gamma + 1.7764\kappa \right\}.$$

Note that α' is still linear for h. Therefore, even if the game definition changes, the adversary's choice of the characteristic function h remains a threshold function h_q for some $0 \le q < 1$. Given that there is a positive probability of sampling γ between 1.6 and 2, it is possible to achieve a lower cost.

▶ **Lemma 23.** *Let $\mu_2(\gamma) = (1-\varepsilon_7)\mu_1(\gamma) + \varepsilon_7(1-\kappa_2)D(\gamma - 1)$, where D is a Dirac-delta function. Then the following holds:*

$$\begin{aligned} &\max_{0 \le q < 1} \max\left\{ \int_1^\infty \gamma \mu_2(\gamma)\, d\gamma + 1.11\kappa_2, \int_1^\infty \alpha'(\gamma, h_q)\mu_2(\gamma)\, d\gamma + 1.7764\kappa_2 \right\} \\ &< \max_{0 \le q < 1} \max\left\{ \int_1^\infty \gamma \mu_1(\gamma)\, d\gamma + 1.11\kappa_2, \int_1^\infty \alpha(\gamma, h_q)\mu_1(\gamma)\, d\gamma + 1.7764\kappa_2 \right\} - \frac{\varepsilon_7}{1000}. \end{aligned}$$

The proof appears in the extended version of this paper [14].

Therefore, we present an improved unifactor approximation algorithm, Algorithm 6.

When $C^* \le K_1 F^*$, an inequality $1.11F^* + 1.7764C^* \le 1.487F^* + 1.487C^*$ is satisfied. For the case $C^* > K_1 F^*$, by Lemma 23, Algorithm 6 shows an improved unifactor approximation with $\varepsilon_8 = \frac{\varepsilon_7}{1000} = 2 \times 10^{-45}$.

8 APX-Hardness

In this section, we prove that UFL is APX-Hard in Euclidean spaces. We use the following result from Austrin, Khot, and Safra [2]. For $\rho \in [-1, +1]$ and $\mu \in [0, 1]$, let $\Gamma_\rho(\mu) := \Pr[X \le \Phi^{-1}(\mu) \wedge Y \le \Phi^{-1}(\mu)]$ where X, Y are standard Gaussian random variables with covariance ρ and Φ is the cumulative density function of the standard normal distribution.

Algorithm 6 Overall uni-factor clustering process.

Derive $(\bar{x}, \bar{y})$ from the optimal LP primal solution (x^*, y^*).
Let X be a random variable that takes the value 1 with probability $1-\kappa_2$, and 0 otherwise.

if $C^* \leq K_1 F^*$ **then**
  Execute the $(1.11, 1.7764)$-approximation algorithm.
else if X $= 1$ **then**
  Execute the $(1.11, 1.7764)$-approximation algorithm.
else
  Takes γ from the distribution $\mu_2(\gamma) = (1-\varepsilon_7)\mu_1(\gamma) + \varepsilon_7(1-\kappa_2)D(\gamma-1)$.
  if $1.6 \leq \gamma \leq 2$ **then**
    Proceed with CONN(CUTINTERVAL(G)) (Algorithm 3 and 4).
  else
    Proceed with GREEDY(G).
  end if
  for all cluster centers j **do**
    Open exactly one nearby facility with a probability of $\bar{x}_{ij}$.
  end for
  for all facilities i not close to any cluster center **do**
    Open each facility with a probability of $\bar{y}_i$ independently.
  end for
  Connect each client to the nearest open facility.
end if

▶ **Theorem 24.** *Assuming the Unique Games Conjecture, for any $q \in (0, 1/2)$ and $\varepsilon > 0$, it is NP-hard to, given a graph $G = (V, E)$, distinguish between the following two cases.*

- *(Completeness) G contains an independent set of size $q|V|$.*
- *(Soundness) For any $T \subseteq V$, the number of edges with both endpoints in T is at least $|E| \cdot (\Gamma_{-q/(1-q)}(\mu) - \varepsilon)$ where $\mu = |T|/|V|$.*

Fix an arbitrary $q \in (0, 1/2)$. Without loss of generality, assume $V = [n]$. Also let $m := |E|$. Our UFL instance has V as the set of facilities and E as the set of clients. The ambient Euclidean space is $\mathbb{R}^n$, and let e_i be the ith standard unit vector (i.e., $(e_i)_i = 1$ and $(e_i)_j = 0$ for every $j \neq i$). Then each facility $i \in V$ is located at e_i and each client $(i, j) \in E$ is located at $e_i + e_j$. Finally, let λ be the common facility cost for every $i \in \mathcal{F}$ to be determined. This finishes the description of the UFL instance.

In the completeness case, there is an independent set U of size qn. We open $V \setminus U$. Since $V \setminus U$ is a vertex cover, every client in E has a facility at distance 1, so the total cost is

$$\lambda(1-q)n + m.$$

In the soundness case, consider any solution that opens $S \subseteq V$ and let $T = V \setminus S$ and $\mu = |T|/n$. By the soundness guarantee, at least $m(\Gamma_{-q/(1-q)} - \varepsilon)$ clients do not have a facility at distance 1. Since every client-facility distance is either 1 or $\sqrt{3}$, the total cost is at least

$$\lambda(1-\mu)n + m\big(1 + (\sqrt{3}-1)(\Gamma_{-q/(1-q)}(\mu) - \varepsilon)\big). \tag{2}$$

For fixed q, the function $\Gamma_{-q/(1-q)}(\mu)$ is a strictly convex function of μ, so if we let λ such that

$$-\lambda n + m(\sqrt{3}-1)\frac{d\Gamma_{-q/(1-q)}(\mu)}{d\mu}|_{\mu=q} = 0,$$

then (2) is minimized when when $\mu = q$, which becomes

$$\lambda(1-q)n + m\big(1 + (\sqrt{3}-1)(\Gamma_{-q/(1-q)}(q) - \varepsilon)\big).$$

Furthermore, we can notice that

$$\lambda = \frac{m}{n} \cdot (\sqrt{3}-1)\frac{d\Gamma_{-q/(1-q)}(\mu)}{d\mu}|_{\mu=q} = \Theta(\frac{m}{n}).$$

Then one can see that the optimal value in the soundness case is at least $m(\sqrt{3}-1)(\Gamma_{-q/(1-q)}(q) - \varepsilon)$ larger than the optimal value in the completeness case. For a fixed q, by choosing ε sufficiently small, one can ensure that this excess is at least a δ fraction of the completeness case optimal value for some constant $\delta > 0$, which proves a $(1+\delta)$-hardness of approximation.

9 Conclusion

The most natural open problem is to get an improved approximation for bifactor or unifactor approximation for UFL. Though we show a strict separation between general and Euclidean metrics for bifactor approximation in a certain regime, it is not achieved for all regimes of bifactor or unifactor approximation.

Whereas our algorithm is based on the *primal rounding* approach of [3] and [15], it might be a fruitful research direction to design a variant of the Jain-Mahdian-Saberi algorithm [12] (greedy algorithm analyzed by the dual fitting method) or Jain-Vazirani [13] (primal-dual algorithm) for a further improvement. In particular, as the best unifactor approximation for UFL in both general and Euclidean metrics employ the $(1.11, 1.7764)$-approximation of the JMS algorithm as a black box, improving the JMS algorithm will directly yield a better result for the best unifactor approximation for UFL. The JV algorithm was already improved in Euclidean spaces [1, 10, 4], but they are not enough for UFL.

References

1 Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. *SIAM Journal on Computing*, 49(4):FOCS17–97, 2019. doi:10.1137/18M1171321.

2 Per Austrin, Subhash Khot, and Muli Safra. Inapproximability of vertex cover and independent set in bounded degree graphs. *Theory of Computing*, 7(1):27–43, 2011. doi:10.4086/TOC.2011.V007A003.

3 Jaroslaw Byrka and Karen Aardal. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. *SIAM Journal on Computing*, 39(6):2212–2231, 2010. doi:10.1137/070708901.

4 Vincent Cohen-Addad, Hossein Esfandiari, Vahab Mirrokni, and Shyam Narayanan. Improved approximations for euclidean k-means and k-median, via nested quasi-independent sets. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1621–1628, 2022. doi:10.1145/3519935.3520011.

5 Vincent Cohen-Addad, Andreas Emil Feldmann, and David Saulpic. Near-linear time approximation schemes for clustering in doubling metrics. *Journal of the ACM (JACM)*, 68(6):1–34, 2021. doi:10.1145/3477541.

6 Vincent Cohen-Addad and CS Karthik. Inapproximability of clustering in lp metrics. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 519–539. IEEE, 2019.

7 Vincent Cohen-Addad, Karthik C S, and Euiwoong Lee. Johnson coverage hypothesis: Inapproximability of k-means and k-median in ℓ_p-metrics. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1493–1530. SIAM, 2022. doi:10.1137/1.9781611977073.63.

8 Vincent Cohen-Addad Viallat, Fabrizio Grandoni, Euiwoong Lee, and Chris Schwiegelshohn. Breaching the 2 lmp approximation barrier for facility location with applications to k-median. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 940–986. SIAM, 2023.

9 Kishen N Gowda, Thomas Pensyl, Aravind Srinivasan, and Khoa Trinh. Improved bi-point rounding algorithms and a golden barrier for k-median. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 987–1011. SIAM, 2023. doi:10.1137/1.9781611977554.CH38.

10 Fabrizio Grandoni, Rafail Ostrovsky, Yuval Rabani, Leonard J Schulman, and Rakesh Venkat. A refined approximation for euclidean k-means. *Information Processing Letters*, 176:106251, 2022. doi:10.1016/J.IPL.2022.106251.

11 Sudipto Guha and Samir Khuller. Greedy strikes back: Improved facility location algorithms. *Journal of algorithms*, 31(1):228–248, 1999. doi:10.1006/JAGM.1998.0993.

12 Kamal Jain, Mohammad Mahdian, and Amin Saberi. A new greedy approach for facility location problems. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 731–740, 2002. doi:10.1145/509907.510012.

13 Kamal Jain and Vijay V Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of the ACM (JACM)*, 48(2):274–296, 2001. doi:10.1145/375827.375845.

14 Euiwoong Lee and Kijun Shin. Facility location on high-dimensional euclidean spaces, 2024. arXiv:2501.18105.

15 Shi Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. *Information and Computation*, 222:45–58, 2013. doi:10.1016/J.IC.2012.01.007.

16 Mohammad Mahdian, Yinyu Ye, and Jiawei Zhang. A 1.52-approximation algorithm for the uncapacitated facility location problem. In *Proc. of APPROX*, pages 229–242, 2002.

17 Robert Alexander Rankin. The closest packing of spherical caps in n dimensions. *Glasgow Mathematical Journal*, 2(3):139–144, 1955.

Toward the Impossibility of Perfect Complete Quantum PKE from OWFs

Longcheng Li
State Key Lab of Processors, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China

Qian Li
Shenzhen International Center for Industrial and Applied Mathematics,
Shenzhen Research Institute of Big Data, China

Xingjian Li
Tsinghua University, Beijing, China

Qipeng Liu
University of California, San Diego, La Jolla, CA, USA

Abstract

In this paper, we study the impossibility of constructing perfect complete quantum public key encryption (QPKE) from quantumly secure one-way functions (OWFs) in a black-box manner. We show that this problem is connected to a fundamental conjecture about the roots of low-degree polynomials on the Boolean hypercube. Informally, the conjecture asserts that for every nonconstant low-degree polynomial, there exists a universal (randomized) way to modify a small number of input bits such that, for every input string, the polynomial evaluated on the modified input string avoids 0 with sufficiently large probability (over the choice of how the input string is modified). Assuming this conjecture, we demonstrate the impossibility of constructing QPKE from quantumly secure one-way functions in a black-box manner, by employing the information-theoretical approach recently developed by Li, Li, Li, and Liu (CRYPTO'24). Towards resolving this conjecture, we provide various pieces of evidence supporting it and prove some special cases. In particular, we fully rule out perfect QPKE from OWFs when the key generation algorithm only makes a logarithmic number of quantum queries, improving the previous work, which can only handle classical queries.

2012 ACM Subject Classification Theory of computation → Cryptographic primitives; Theory of computation → Quantum complexity theory

Keywords and phrases Qautnum public-key encryption, Boolean function analysis

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.71

Funding *Qian Li*: Qian Li's work was supported by Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone Project (No.HZQSWS-KCCYB-2024016).

1 Introduction

Public-key encryption (PKE) is a fundamental primitive in modern cryptography. It enables secure communication through an insecure channel. A server generates two keys, the secret key and the public key; the secret key is owned only by the server, whereas the public key is broadcasted to everyone else. PKE allows everyone to send messages to the server securely; even if a malicious party, who keeps listening to the channel, has no idea about the actual messages. Its counterpart, secret-key encryption, requires pre-shared keys to conduct secure


© Longcheng Li, Qian Li, Xingjian Li, and Qipeng Liu;
licensed under Creative Commons License CC-BY 4.0

communication. Since the first proposal [1] by Diffie and Hellman [13], PKE has become one of the most important cryptographic primitives and concepts, with impacts ranging from theoretical computer science to real-world constructions.

Despite enjoying all the advancements in secret-key encryption (SKE), all PKE schemes have more structures. Therefore, PKE seems to require strictly stronger assumptions than one-way functions. This observation was later confirmed by Impagliazzo and Rudich [16]: there was no black-box construction of PKE solely from one-way functions (or in the random oracle model).

Quantum information has changed the landscape of cryptography, especially weakening the assumptions needed for various cryptographic primitives; for example, quantum key distribution [10], oblivious transfer/multi-party computation [7, 15], commitment [3, 23]. Talking about PKE, first introduced by Morimae and Yamakawa [22], then followed by [12, 17, 19, 6], QPKE with either quantum keys or quantum ciphertext can be constructed from OWFs. However, quantum public keys and quantum ciphertext are often more difficult to broadcast, use and authenticate than their classical counterparts. Thus, the possibility of basing quantum PKE with classical public key, secret key and ciphertext on one-way functions is still intriguing and largely unanswered. In the rest of the work, we will simply denote such QPKE with classical keys and ciphertext by "QPKE". Following the similar questions in [16], in this work, we are interested in the problem:

Question 1: Whether QPKE can be constructed solely in the quantum random oracle model (QROM)?

The direction was investigated first by Austrin, Chung, Chung, Fu, Lin and Mahmoody [5] and later by Li, Li, Li and Liu [18]. Austrin et al. proposed the so-called "polynomial compatible conjecture" (or PCC for short); basing on PCC, they fully rule out QPKE from OWFs. However, their conjecture is tailored to this separation problem; to which we will compare our conjecture later on. Besides, they are able to prove the non-existence of perfect complete QPKE in the QROM when (i) Enc or (ii) both Gen and Dec make only classical queries to a random oracle. Li et al. improved the result (ii) and showed that as long as the key generation algorithm Gen only makes classical queries, perfect correct QPKE can not exist in the QROM.

1.1 Our Results

In this work, we show that Question 1 is closely related to a fundamental conjecture about the roots of low-degree polynomials on the Boolean hypercube. More specifically, we propose the following conjecture, and show that it implies a full impossibility result of perfect complete QPKE in the QROM.

▶ **Conjecture 1.** *For any nonconstant function $f : \{-1,1\}^n \to \mathbb{R}$ with degree d, the following holds. There exists a distribution $\mathcal{D}$ on the set of partial assignments ρ with $|\rho| \leq \mathrm{poly}(d)$ such that: for any $x \in \{-1,1\}^n$,*

$$\Pr_{\rho\sim\mathcal{D}}[f(x^\rho) \neq 0] \geq 1/\mathrm{poly}(d).$$

[1] The UK Government Communications Headquarters proposed and developed the same scheme even earlier than Diffie and Hellman (1973 and 1974), and was later declassified by the British government in 1997.

A partial assignment is denoted by $\rho \in \{-1, 1, \star\}^n$ and $|\rho|$ is defined as the number of non-$\star$ entries. x^ρ is defined as an input string whose i-th bit will be equal to x_i if $\rho_i = \star$, and otherwise equal to ρ_i. In other words, the conjecture asserts that for every degree d nonconstant f, there is a universal (randomized) way to modify $\mathrm{poly}(d)$ bits such that, every input string x has an inverse-polynomial probability of evaluating to non-zero under this randomized modification.

▶ Remark 2 (Comparing PCC in [5] with our conjecture). PCC (Conjecture 1.2) roughly asserts that, for two distributions F, G over low degree polynomials with bounded influences, there always exists $f \in F, g \in G$ and x such that $f(x)g(x) \neq 0$. To the best of our knowledge, there is no obvious connection between PCC and our conjecture. Nonetheless, we hold that our conjecture appears simpler and offers an alternative path towards the ultimate objective; since our conjecture only involves a single polynomial (instead of two distributions) and resembles many fundamental notions in the literature of boolean function analysis. Furthermore, PCC might be too strong to hold: as it will imply attacks with only a polynomial number of classical queries, even if both Alice and Bob make quantum queries; on the other hand, our Eve makes quantum queries.

With the conjecture, we now present our main theorem.

▶ **Theorem 3** (Main Theorem). *If Conjecture 1 is true, then perfect complete QPKE does not exist in the QROM.*

▶ Remark 4. In the proof of Theorem 3, f is treated as a probability and ranges in $[0, 1]$. However, it would not weaken Conjecture 1 if we restrict the range of f to $[0, 1]$. Precisely, in Conjecture 1, it does not lose generality to assume that f is nonnegative, because we can consider f^2 instead; then we can scale the function to fit in the range $[0, 1]$.

Conjecture 1 is closely related to notions like certificate complexity, sensitivity in the literature of boolean function analysis (BFA), and also to the celebrated Combinatorial Nullstellensatz of Alon [2], making it a considerably natural conjecture. With the tools in BFA and the Combinatorial Nullstellensatz, we can provide many pieces of evidence and prove special cases. We mention some of the most important results below while leaving others in the main body.

Special Case 1: Gen makes only classical queries

This is the case that already was analyzed in [18]; there, they did not go through BFA, but rather proved it directly. We demonstrate the versatility of our conjecture and framework, by showing:

▶ **Lemma 5.** *Conjecture 1 holds for all f that can be expressed as the acceptance probability of a (randomized) classical decision tree.*

▶ **Corollary 6** (Recasting [18]). *Perfect complete QPKE does not exist in the QROM, if* Gen *only makes classical queries.*

Evidence 2: The Combinatorial Nullstellensatz

The following relaxed version of Conjecture 1 directly follows from the Combinatorial Nullstellensatz of Alon (Lemma 22).

▶ **Lemma 7.** *For any nonconstant function $f : \{-1,1\}^n \to \mathbb{R}$ with degree d, the following holds. There exists a distribution $\mathcal{D}$ on the set of partial assignments ρ with $|\rho| \leq d$ such that: for any x,*

$$\Pr_{\rho\sim\mathcal{D}}[f(x^\rho) \neq 0] \geq 1/2^d.$$

Somewhat surprisingly, we find that if the probability of $f(x^\rho)$ being non-zero in Lemma 7 can be improved from $1/2^d$ to $1/2^{d^c}$ for an arbitrary $c < 1$, then Conjecture 1 would be affirmed. Formally, the following conjecture is equivalent to Conjecture 1:

▶ **Conjecture 8.** *There is a universal constant $c < 1$, such that for any nonconstant function $f : \{-1,1\}^n \to \mathbb{R}$ with degree d, the following holds. There exists a distribution $\mathcal{D}$ on the set of partial assignments ρ with $|\rho| \leq \mathrm{poly}(d)$ such that: for any x,*

$$\Pr_{\rho\sim\mathcal{D}}[f(x^\rho) \neq 0] \geq 1/2^{d^c}.$$

▶ **Lemma 9.** *Conjecture 8 holds if and only if Conjecture 1 holds.*

Special Case 3: Gen makes $O(\log n)$ quantum queries

With Lemma 7, by following a similar argument in [18], we can directly conclude that when the generation algorithm only makes $O(\log n)$ quantum queries, such QPKE does not exist in the QROM.

▶ **Lemma 10.** *Perfect complete QPKE does not exist in the QROM, if* Gen *only makes $O(\log n)$ quantum queries.*

Thus, we make a step forward by improving the result (ii) in [5] and that in [18].

Table 1 Provable separations in [5, 18] and this work. Q and C denote a polynomial number of quantum and classical queries, respectively.

	[5] (i)	[5] (ii)	[18]	This work
Gen	Q	C	C	log Q
Enc	C	Q	Q	Q
Dec	Q	C	Q	Q

Special Case 4: Gen has "uniform" outputs

Here, by **"uniform"**, we mean that the output of Gen^H has support of the same size for every possible oracle H, and the output distribution is uniform over the support. In other words, $|\mathsf{supp}(\mathsf{Gen}^H)| = |\mathsf{supp}(\mathsf{Gen}^{H'})|$ for every pair of H, H', and Gen^H is uniform over samples with a non-zero probability. To resolve this case, we prove the following special case of Conjecture 1:

▶ **Lemma 11.** *Conjecture 1 holds for all f that takes at most $\mathrm{poly}(d)$ values.*

▶ **Corollary 12.** *Perfect complete QPKE does not exist in the QROM, if* Gen *is "uniform".*

Evidence 5: Other equivalent conjectures

By the min-max principle, Conjecture 1 has the following equivalent form.

▶ **Conjecture 13.** *For any nonconstant function $f : \{-1,1\}^n \to \mathbb{R}$ with degree d, and any distribution X on $\{-1,1\}^n$, there exists a partial assignment with $|\rho| \leq \mathrm{poly}(d)$ such that $\Pr_{x\sim X}[f(x^\rho) \neq 0] \geq 1/\mathrm{poly}(d)$.*

▶ **Lemma 14.** *Conjecture 13 holds if and only if Conjecture 1 holds. Thus, if Conjecture 13 is true, then perfect complete QPKE does not exist in the QROM.*

In the main body, we provide additional evidence for Conjecture 13 (and consequently for Conjecture 1). Conjecture 13 has the advantage of requiring only a single partial assignment that works for the given input distribution, rather than a distribution of partial assignments. This simplifies the reasoning considerably, and much of our evidence supports Conjecture 13.

1.2 Techniques

Here, we give a brief overview of why Conjecture 1 implies a full impossibility result of QPKE in the QROM and the main differences between [18] and this work.

We will be mostly focusing on key agreement protocols in the QROM. In the QROM, all parties can quantumly query a random oracle. A key agreement protocol is an interactive protocol between two query-efficient quantum algorithms Alice and Bob, exchanging classical messages. Their goal is to agree on a key, whereas any query-efficient adversary can not learn the key, even observing all the communication on this channel. It is easy to see that QPKE implies a two-round key agreement protocol:

- Alice sends a single message m_0 to Bob; in this case, m_0 will be the public key.
- Bob sends a message m_1 to Alice; in this case, m_1 will be the encryption of a random key (which will be the agreed key in the two-round key agreement protocol).
- Finally, they both output their own keys k_A, k_B; in this case, Alice decrypts the ciphertext using her secret key, whereas Bob simply outputs the random key.

Thus, to rule out QPKE in the QROM, it is sufficient to rule out the two-round key agreement in the QROM.

Li, Li, Li and Liu [18] proposed the following framework that rules out QPKE with classical-query key generation in the QROM. Let $\mathsf{View}_A, \mathsf{View}_B$ denote the view of Alice and Bob, right after m_1 is received by Alice. They showed that, based on a novel approach called approximate quantum Markov chain, a quantum-query Eve can reconstruct Alice's view View'_A such that

$$\mathsf{View}'_A\mathsf{View}_B \approx \mathsf{View}_A\mathsf{View}_B.$$

More precisely, Eve can reconstruct Alice's register such that the margin of the fake Alice and the real Bob is arbitrarily close to the real Alice and the real Bob. However, this is not sufficient; since to compute the key k_A, Alice potentially needs to perform computation depending on its current state, the random oracle and all the messages m_0, m_1. As the marginal is defined by tracing out the random oracle, we do not know what oracle to work with. For example, a direct attempt is to run fake Alice under the real random oracle H; but it is possible that under real H, conditioned on the messages being m_0, m_1, the view of Alice and Bob will never equal to $\mathsf{View}'_A\mathsf{View}_B$, making the attempt meaningless.

Li et al. solved this by adapting Bob; while perfectly preserving the functionality of the key agreement protocol, the new Bob has one additional nice property (let us call it "stability"):

- if under a random oracle H, input message m_0, Bob outputs certain m_1 with non-zero probability
- then for every H' that has a small Hamming distance to H[2], Bob on H' and m_0 still has non-zero probability to output m_1.

With this property, they show that the two-round key agreement can not exist, when m_0 together with View_A can be computed by a classical decision tree (or Gen only makes classical queries). For any fake Alice view View'_A, it depends on at most d locations of a random oracle; here d is the number of queries made by the classical decision tree. Thus, for any oracle H_0, as long as it is consistent on these d locations, Alice on H_0 can output View'_A and m_0 with non-zero probability. They construct an oracle H' such that: (i) it is almost H, except (ii) on those d locations, H' is reprogrammed to be consistent with these d locations.

It is clear that Alice on H' still outputs m_0, View'_A with a non-zero probability (consistency of these d locations). Moreover, the "stability" ensures that the adapted Bob on H' and m_0 outputs View_B also with a non-zero probability. Thus, there is a non-zero probability, under the oracle H', Alice and Bob will end up with $\mathsf{View}'_A, \mathsf{View}_B$ and transcripts (m_0, m_1). Since we assume the QPKE is perfect, after the whole execution, Alice and Bob should agree on a key. As for Eve, it simply runs Alice with View'_A, m_1 on H' and will recover the key.

The above reasoning in [18] does not work even if Alice only makes a single quantum query, and thus they can only rule out QPKE when Gen makes classical queries. Since a single quantum query can "store" information about exponentially many inputs, it seems that fixing a smaller number of input-output behaviors does not fix Alice's behavior.

We realize that, we do not need to keep Alice's behavior the same; we are only required to keep these probabilities non-zero. More precisely, our solution is to directly look at the polynomial f describing the probability that a quantum Alice on some oracle, outputs (m_0, View'_A),

$$f(x) = \Pr[(m_0, \mathsf{View}'_A) \leftarrow A^x],$$

where we treat x as the random oracle fed into A. As the random oracle will be the input to a polynomial, to be consistent with the literature of BFA, we will denote a random oracle by x. Since A only makes d quantum queries, such a polynomial f has degree at most $2d$. The real random oracle is some x of which we have no knowledge about; the goal is to find another x' such that

- $f(x') > 0$, meaning Alice on the random oracle x' outputs View'_A, m_0 with a non-zero probability, just like the classical case; and,
- x' and x have a small Hamming distance, so that Bob still has non-zero probability to output View_B, m_1.

As we do not know x, the only way we can obtain x' is by locally modifying some locations of x. For example, reprogramming some locations of the real oracle, and running the rest of Alice under the reprogrammed oracle – this is where the partial assignment takes place. If we can find a partial assignment ρ with $|\rho|$ is small, such that for every x, $f(x^\rho) > 0$, then the problem is resolved! This is the high-level intuition why our conjecture (Conjecture 1) implies a full impossibility result of QPKE in the QROM.

[2] We ignore a detail here: H' should be consistent with H on a small set of inputs, whereas on other inputs, their Hamming distance is small. This does not change the reasoning, thus we make the simplification in the introduction.

Towards Conjecture 1, we remark that the Combinatorial Nullstellensatz [2] directly implies the existence of such ρ. More specifically, the Combinatorial Nullstellensatz claims that: given any maximal monomial x_S of f, for any $x \in \{-1,1\}^n$, there exists a ρ with $\mathsf{supp}(\rho) = S$ such that $f(x^\rho) \neq 0$. Therefore, Conjecture 1 holds if we swap the order of quantifiers of ρ and x. Consequently, by letting $\mathcal{D}$ be the uniform distribution on the set of partial assignments ρ with $\mathsf{supp}(\rho) = S$, we have $\Pr_{\rho\sim\mathcal{D}}[f(x^\rho) \neq 0] \geq 1/2^d$ for any x, as claimed by Lemma 7. In particular, when Gen makes only $O(\log n)$ queries and then $\deg(f) = O(\log n)$, we have $\Pr_{\rho\sim\mathcal{D}}[f(x^\rho) \neq 0] \geq 1/\mathrm{poly}(n)$, which implies Lemma 10.

1.3 Related works

In Conjecture 1, we characterize d-query quantum algorithms as degree-$2d$ polynomials [8]. This characterization was further strengthened by Aaronson and Ambainis [1] in terms of bounded degree-$2d$ block-multilinear polynomials. Later, Arunachalam, Briet, and Palazuelos [4] provided an exact characterization of quantum query algorithms in terms of the so-called completely bounded norm.

2 Preliminaries

2.1 Basic notions in Boolean function analysis

Every function $f : \{-1,1\}^n \to \mathbb{R}$ on the hypercube can be uniquely expressed as a multilinear polynomial

$$f(x) = \sum_{S\subseteq[n]} a_S \cdot x_S,$$

where $x_S := \Pi_{i\in S} x_i$. Indeed, this expression is called the *Fourier expansion* of f, where $a_S = 2^{-n}\sum_x f(x)\cdot x_S$. The *degree* of f, denoted $\deg(f)$, is defined as the degree of its multilinear polynomial expression, i.e., $\max\{|S| \mid a_S \neq 0\}$. A monomial x_S is called *maximal* if it has degree $\deg(f)$, i.e., $|S| = \deg(f)$ and $a_S \neq 0$. The *rank* of f, denoted $\mathrm{rank}(f)$, is the maximum number of disjoint maximal monomials.

A *partial assignment* is a function $\rho : [n] \to \{-1, 1, \star\}$. We define the support of ρ as $\mathsf{supp}(\rho) := \{i|\rho_i \neq \star\}$, and the size as $|\rho| := |\mathsf{supp}(\rho)|$. For $x \in \{-1,1\}^n$, we define the modification of x with ρ, denoted x^ρ, as the string $x' \in \{-1,1\}^n$ where $x'_i = \rho_i$ for $i \in \mathsf{supp}(\rho)$ and $x'_i = x_i$ for any other i.

We use $\mathsf{Var}(f) := \mathbb{E}_x[f(x)^2] - \big(\mathbb{E}_x f(x)\big)^2$ to denote f's variance, and define $\|f\|_\infty := \max_x |f(x)|$. A real polynomial p ϵ-approximates f if $|f(x)-p(x)| \leq \epsilon$ for every $x \in \{-1,1\}^n$. The *approximate degree* of f, denoted by $\widetilde{\deg}(f)$, is defined to be the minimum degree needed to 1/3-approximate f.

The following lemma will be used.

▶ **Lemma 15** ([8]). *Suppose a quantum algorithm makes d queries to a Boolean string $x \in \{-1,1\}^n$, and the acceptance probability is denoted by $f(x)$. Then the function $f : \{-1,1\}^n \to \mathbb{R}$ has degree at most $2d$. That is, f can be expressed as*

$$f(x) = \sum_{|S|\leq 2d} a_S \cdot x_S$$

2.2 Quantum public key encryption

In this paper, we will consider constructions in the quantum random oracle model. Given security parameter λ, the oracle H is chosen from the uniformly random distribution over the family of functions $\mathcal{H}_\lambda \colon [2^{n_\lambda}] \to \{0,1\}$, where n_λ is a polynomial of λ. The quantum circuit has access to the oracle unitary U_H that maps $|i,y\rangle$ to $|i, y \oplus H(i)\rangle$. We can also view the oracle unitary in the phase basis $U'_H |i,y\rangle \to (-1)^{H(i)y} |i,y\rangle$.

We will consider quantum public key encryption with classical public key pk and ciphertext ct in this paper. Particularly, we will consider the quantum public key encryption (QPKE) scheme in the quantum random oracle model (QROM) defined as follows:

▶ **Definition 16** (Quantum public key encryption in QROM)**.** *A public key encryption scheme, relative to a random oracle $H \leftarrow \mathcal{H}_\lambda$ consists of the following three bounded quantum query algorithms:*

- $\mathsf{Gen}^H(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$*: The key generation algorithm that generates a pair of classical public key* pk *and secret key* sk*.*
- $\mathsf{Enc}^H(\mathsf{pk}, m) \to \mathsf{ct}$*: the encryption algorithm that takes a public key* pk*, the plaintext m, produces the ciphertext* ct*.*
- $\mathsf{Dec}^H(\mathsf{sk}, \mathsf{ct}) \to m'$*: the decryption algorithm that takes secret key* sk *and ciphertext* ct *and outputs the plaintext m'.*

The algorithms should satisfy the following requirements:

Perfect Completeness $\Pr\left[\mathsf{Dec}^H\left(\mathsf{sk}, \mathsf{Enc}^H(\mathsf{pk}, m)\right) = m \colon \mathsf{Gen}^H(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})\right] = 1.$

IND-CPA Security For any QPT adversary $\mathcal{E}^H$, for every two plaintexts $m_0 \neq m_1$ chosen by $\mathcal{E}^H(\mathsf{pk})$ we have

$$\Pr\left[\mathcal{E}^H\left(\mathsf{pk}, \mathsf{Enc}^H(\mathsf{pk}, m_b)\right) = b\right] \leq \frac{1}{2} + \frac{\epsilon(\lambda)}{2}.$$

where we call $\epsilon(\lambda)$ as the advantage of the adversary. The scheme is IND-CPA secure if $\epsilon(\lambda)$ is negligible for any adversary $\mathcal{E}^H$.

3 Consequences of our BFA conjecture

In this section, we will show the consequence in cryptography of Conjecture 1. If Conjecture 1 holds, we can obtain a black box separation between post-quantum secure one-way functions and quantum public key encryption schemes.

▶ **Theorem 17** (Restate of Theorem 3)**.** *Assuming Conjecture 1 is true, for any quantum public key encryption scheme in the quantum random oracle model, if* Gen^H *and* Enc^H *make d queries to H in total, and* Dec^H *makes D queries, there exists an adversary Eve $\mathcal{E}^H$ which can break the IND-CPA security with advantage $1/\mathrm{poly}(d)$ by making $O(\mathrm{poly}(d) + D)$ queries.*

It is known that we can use a QPKE scheme for a two-message key agreement protocol, by setting the first message $m_0 = \mathsf{pk}$, and the second message $m_1 = \mathsf{Enc}(\mathsf{pk}, k)$, where k is the key the two parties agree on. We can assume the key k is chosen uniformly random from $\{0,1\}^n$. In the following section, we will refer to the two parties as $\mathcal{A}$ and $\mathcal{B}$, and call the algorithm of $\mathcal{A}$ before sending m_0 as $\mathcal{A}_0$, and the algorithm after receiving message m_1 as $\mathcal{A}_1$. If we can learn the key k with probability p, we can also break the semantic security of the QPKE scheme with advantage p, which implies that we also break the IND-CPA security of the protocol. In the language of key agreement, we are considering the case where $\mathcal{A}_0, \mathcal{A}_1$, and $\mathcal{B}$ can both make quantum queries while only sending classical messages, and we will show how to learn the key k with probability $1/\mathrm{poly}(d)$.

Let us recall the results from [18]. In their paper, they are considering the case where $\mathcal{A}_0$ can only make classical queries. The key part of their analysis is to show how to construct a register A', such that there exists some oracle H', the content View'_A in A' is consistent with H' and transcript $\pi = (m_0, m_1)$. We can think the algorithm $\mathcal{B}$ proceeds as follows: it first receives the message m_0 from $\mathcal{A}$, and after making some queries to the oracle H, it makes a measurement in the computational basis, and generates the key k_B and the second message m_1. The state under consideration in [18] $\sigma_{AB} = \mathbb{E}_H[\sigma_{AB}^H]$ is the state before $\mathcal{B}$ makes the final measurement and sends the second message m_1, where σ_{AB}^H is the state under oracle H, and the expectation is over the posterior distribution of H given the first message m_0.

To construct the register A', they used tools from quantum information theory. They used the following theorem from [14].

▶ **Theorem 18.** *For any state ρ_{AEB} over systems AEB, there exists a channel $\mathcal{T} : \mathsf{E} \to \mathsf{E} \otimes \mathsf{B}'$ such that the trace distance between the reconstructed state $\sigma_{\mathsf{A}'\mathsf{E}'\mathsf{B}'} = \mathcal{T}(\rho_{\mathsf{AE}})$ and the original state ρ_{AEB} is at most*

$$\sqrt{\ln 2 \cdot I(\mathsf{A} : \mathsf{B}|\mathsf{E})_\rho}.$$

The theorem says that if the conditional mutual information $I(\mathsf{A} : \mathsf{B}|\mathsf{E})$ is small, we can generate a consistent copy of A' by a channel only acting on E.

To decrease the conditional mutual information $I(\mathsf{A} : \mathsf{B}|\mathsf{E})$, we have the following lemma from [18].

▶ **Definition 19** (Permutation Invariance)**.** *Let $\mathsf{A}_1, \mathsf{A}_2, \mathsf{A}_3, \ldots, \mathsf{A}_t, \mathsf{B}$ be $(t+1)$-partite quantum system. Given the joint state $\rho_{\mathsf{BA}_1\mathsf{A}_2\cdots\mathsf{A}_t}$, we say $A_1, \ldots, A_t$ are permutation invariant, if for any permutation π on $[t]$, we have*

$$\rho_{\mathsf{BA}_1\mathsf{A}_2\cdots\mathsf{A}_t} = \rho_{\mathsf{BA}_{\pi(1)}\mathsf{A}_{\pi(2)}\cdots\mathsf{A}_{\pi(t)}}.$$

▶ **Lemma 20.** *Let $\mathsf{A}_1, \mathsf{A}_2, \mathsf{A}_3, \ldots, \mathsf{A}_t, \mathsf{B}, \mathsf{E}$ be $(t+2)$-partite quantum system. Suppose the state of the composite system $\rho_{\mathsf{BEA}_1\mathsf{A}_2\cdots\mathsf{A}_t}$ is fully separable. If $\mathsf{A}_1, \mathsf{A}_2, \mathsf{A}_3, \ldots, \mathsf{A}_t$ are permutation invariant, then there is a $0 \le i \le t-1$ such that*

$$I(\mathsf{A}_t : \mathsf{B} \mid \mathsf{E}, \mathsf{A}_1, \ldots, \mathsf{A}_i)_\rho \le S(\mathsf{B})/t.$$

The lemma shows that if Eve parallel repeats multiple copies of $\mathcal{B}$, the mutual information will be decreased.

The following lemma from [9] will also be used.

▶ **Lemma 21.** *Consider a quantum algorithm $\mathcal{B}$ that makes d queries to an oracle H. Denote the quantum state immediately after t queries to the oracle as*

$$|\psi_t\rangle = \sum_{x,w} \alpha_{x,w,t} |x, w\rangle,$$

where w is the content of the workspace register. Denote the query weight q_x of input x as

$$q_x = \sum_{t=1}^{d} \sum_{w} |\alpha_{x,w,t}|^2.$$

For any oracle $\tilde{H}$, denote $|\phi_d\rangle$ as the final state before measurement obtained by running $\mathcal{B}$ with oracle $\tilde{H}$, we have that

$$\||\psi_d\rangle - |\phi_d\rangle\| \le 2\sqrt{d} \sqrt{\sum_{x:\, \tilde{H}(x) \neq H(x)} q_x}.$$

In [18], they defined the heavy query of Bob as $\{x\colon q_x \geq \epsilon^2/d^2\}$. We summarize their adversary Eve's algorithm $\mathcal{E}$ as follows:

1. Eve parallelly simulates Bob's algorithm $\mathcal{B}^H(m_0)$ for multiple times. In this process, Eve records heavy queries of $\mathcal{B}$ under H classically and maintains an input-output pair register $R_E = \{(i_E, H(i_E))\}$.
2. By Lemma 20, if Eve repeat $\mathcal{B}^H(m_0)$ for $\mathrm{poly}(d, 1/\epsilon)$ times, the conditional mutual information $I(\mathsf{A} : \mathsf{B} \mid \mathsf{E})$ of state σ_{ABE} can be reduced to smaller than $O(\epsilon^2)$. By Theorem 18,the channel $\mathcal{T}\colon \mathsf{E} \to \mathsf{A'E}$ provides a state $\mathcal{T}(\sigma_{BE}) = \sigma_{A'BE}$ is ϵ close to the state σ_{ABE}. In the following section, we will choose $\epsilon = 1/\mathrm{poly}(d)$.
3. Eve measures the secret key register as well as the query input-output history register of $\mathsf{A'}$. Note that in their case, we can assume $\mathcal{A}_0^H$ records its classical queries. The measurement will give us some secret key $\mathsf{sk'}$ and input-output pairs $R_{A'} = \{(i_{A'}, H'(i_{A'}))\}$, since $\sigma_{A'BE} \approx_\epsilon \sigma_{ABE}$, with high probability, the measurement result $\mathsf{sk'}$ and $R_{A'}$ are consistent with message m_0 and R_E. Specifically, every query that is both in $R_{A'}$ and R_E should be consistent, meaning $R_{A'}$ is consistent with $\mathcal{B}$'s heavy queries under oracle H.
4. Finally, by the observation beyond, if the oracle H is reprogrammed to H' using the input-output pairs in $R_{A'}$, from $\mathcal{B}$'s view, only $\mathrm{poly}(d)$ light query points under H are modified. Using Lemma 21, it can be shown that w.h.p., $\mathcal{B}^{H'}(m_0)$ will output m_1 with nonzero probability, implying that $\pi = (m_0, m_1)$ is a valid transcript under oracle H'. Thus by perfect completeness, if we simulate $\mathcal{A}_1$ given input $(\mathsf{sk'}, m_1)$ over oracle H', it can still agree with $\mathcal{B}$.

We refer interested readers to their paper for the proof details.

Our observation is that in step 3, we do not need to generate the input-output pair $R_{A'}$ by measuring $\mathsf{A'}$. Given the algorithm of $\mathcal{A}_0$, if we can find some $\mathrm{poly}(d)$-sized assignment $R_{A'}$ that is consistent with R_E and guarantees the output probability of $(\mathsf{sk'}, m_0)$ is still non-zero, by similar arguments, we can see that $\mathcal{B}$ is consistent with the reprogrammed oracle H', and will output m_1 with non-zero probability. From Lemma 15, if we define the algorithm outputs $(\mathsf{sk'}, m_0)$ as acceptance, we can characterize the probability with a $2d$-degree polynomial $f(x)$, viewing the truth table of random oracle H as a $N_\lambda = 2^{n_\lambda}$ length vector x.

Now we show how to leverage Conjecture 1 to prove our main theorem. We use x for the truth table of original oracle H, setting $x_i = (-1)^{H(i)}$, and ρ_E for the restriction given by R_E, and apply Conjecture 1 to polynomial $g(x) = f(x^{\rho_E})$. In an operational meaning, considering $g(x)$ means that we first fix the input-output pairs given by R_E when selecting the random oracle. We now argue $g(x)$ is a non-zero polynomial with high probability. We can obtain the joint view $\mathsf{View}_{A'E} = (\mathsf{sk'}, m_0, R_E)$ of $\mathsf{A'E}$ by performing a computational basis measurement on corresponding registers. Since $\sigma_{A'E}$ is ϵ close to σ_{AE}, we can see that w.p. $1 - \epsilon$, $\mathsf{View}_{A'E} = (\mathsf{sk'}, m_0, R_E)$ is a possible valid view obtained by measuring σ_{AE}.

By Conjecture 1, there exists a distribution of polynomial-sized restrictions $\mathcal{D}$ such that for any x, $\Pr_{\rho\sim\mathcal{D}}[g(x^\rho) \neq 0] \geq 1/\mathrm{poly}(d)$, with $|\rho| \leq \mathrm{poly}(d)$. In our case, this implies that $\Pr_{\rho\sim\mathcal{D}}[f(x^{\rho\cup\rho_E}) \neq 0] \geq 1/\mathrm{poly}(d)$. If we select a polynomial-sized restriction ρ, there is $1/\mathrm{poly}(d)$ probability such that $(\mathsf{sk'}, m_0)$ is still a valid view of $\mathcal{A}_0$ under the new oracle given by $x^{\rho\cup\rho_E}$. The other arguments follow from the original proof.

4 Toward our BFA conjecture: evidences and special cases

In this section, we explore Conjecture 1 by providing some evidence and proving some special cases.

4.1 Main evidence: Combinatorial Nullstellensatz

The main evidence for Conjecture 1 comes from the celebrated Combinatorial Nullstellensatz of Alon [2], which implies Conjecture 1, except with $\Pr_{\rho\sim\mathcal{D}}[f(x^\rho) \neq 0] \geq 1/2^d$ instead of $\Pr_{\rho\sim\mathcal{D}}[f(x^\rho) \neq 0] \geq 1/\mathrm{poly}(d)$. Let us state the special case of the Combinatorial Nullstellensatz for polynomials on the hypercube.

▶ **Lemma 22** ([2]). *Let $f : \{-1,1\}^n \to \mathbb{R}$ be any nonconstant function with degree d, and x_S be any maximal monomial of f. For any $x \in \{-1,1\}^n$, there exists a ρ with $\mathsf{supp}(\rho) = S$ such that $f(x^\rho) \neq 0$.*

Consequently, by letting $\mathcal{D}$ be the uniform distribution on the set of partial assignments ρ with $\mathsf{supp}(\rho) = S$, we have $\Pr_{\rho\sim\mathcal{D}}[f(x^\rho) \neq 0] \geq 1/2^d$ for any x.

We remark that Lemma 22 was also proven by Midrijanis [21]. Even though Lemma 22 has an exponential rather than polynomial dependence on $1/d$, we observe that it already has a nontrivial implication on the separation between QPKE and OWFs. Precisely, it implies that

▶ **Theorem 23** (Restate of Lemma 10). *For any quantum public key encryption scheme in the quantum random oracle model, if Enc^H makes d queries to H, Gen^H makes $O(\log d)$ queries, and Dec^H makes D queries, there exists an adversary Eve $\mathcal{E}^H$ which can break the IND-CPA security with advantage $1/\mathrm{poly}(d)$ by making $O(\mathrm{poly}(d) + D)$ queries.*

The proof of Theorem 23 is the same as Theorem 17 by replacing Conjecture 1 with Lemma 22.

4.2 Proof of Lemma 9

In this subsection, we present the proof of Lemma 9, which claims that Conjecture 1 is equivalent to a seemingly much weaker conjecture, namely Conjecture 8.

▶ **Conjecture 8.** *There is a universal constant $c < 1$, such that for any nonconstant function $f : \{-1,1\}^n \to \mathbb{R}$ with degree d, the following holds. There exists a distribution $\mathcal{D}$ on the set of partial assignments ρ with $|\rho| \leq \mathrm{poly}(d)$ such that: for any x,*

$$\Pr_{\rho\sim\mathcal{D}}[f(x^\rho) \neq 0] \geq 1/2^{d^c}.$$

▶ **Lemma 9.** *Conjecture 8 holds if and only if Conjecture 1 holds.*

Proof. Conjecture 1 implying Conjecture 8 is trivial. Conversely, given any nonconstant function $f : \{-1,1\}^n \to \mathbb{R}$ with degree d, define

$$\tilde{f}(x_1, \ldots, x_t) := \prod_{i=1}^{t} f(x_i)$$

where $x_i \in \{-1,1\}^n$. Note that $\deg(\tilde{f}) = td$. Assuming Conjecture 8 holds for $\tilde{f}$, there exists a distribution $\tilde{\mathcal{D}}_0$ of partial assignment $\tilde{\rho} = (\rho_1, \ldots, \rho_t) \in (\{-1,1,\star\}^n)^t$ with $|\tilde{\rho}| \leq (dt)^{c_1}$ such that

$$\min_{\tilde{x}} \Pr_{\tilde{\rho}\sim\tilde{\mathcal{D}}_0}\left[\tilde{f}(\tilde{x}^{\tilde{\rho}}) \neq 0\right] \geq \frac{1}{2^{(dt)^{c_2}}}$$

where c_1, c_2 are universal constants satisfying $c_1 \geq 0$ and $0 \leq c_2 < 1$. Observe that

$$\max_{\tilde{\mathcal{D}}} \min_{\tilde{x}} \Pr_{\tilde{\rho}\sim\tilde{\mathcal{D}}}\left[\tilde{f}(\tilde{x}^{\tilde{\rho}}) \neq 0\right] \geq \min_{\tilde{x}} \Pr_{\tilde{\rho}\sim\tilde{\mathcal{D}}_0}\left[\tilde{f}(\tilde{x}^{\tilde{\rho}}) \neq 0\right] \geq \frac{1}{2^{(dt)^{c_2}}}$$

where $\tilde{\mathcal{D}}$ is a distribution of partial assignment $\tilde{\rho} = (\rho_1, \ldots, \rho_t) \in (\{-1, 1, \star\}^n)^t$ with $|\rho_i| \leq (dt)^{c_1}$ for all i. Then by Lemma 24, we have

$$\left(\max_{\mathcal{D}} \min_{x} \Pr_{\rho\sim\mathcal{D}}[f(x^\rho) \neq 0]\right)^t = \max_{\tilde{\mathcal{D}}} \min_{\tilde{x}} \Pr_{\tilde{\rho}\sim\tilde{\mathcal{D}}}\left[\tilde{f}(\tilde{x}^{\tilde{\rho}}) \neq 0\right] \geq \frac{1}{2^{(dt)^{c_2}}}$$

where $\mathcal{D}$ is a distribution of partial assignment $\rho \in \{-1, 1, \star\}^n$ with $|\rho| \leq (dt)^{c_1}$. Thus there exists a distribution $\mathcal{D}^*$ such that

$$\left(\min_{x} \Pr_{\rho\sim\mathcal{D}^*}[f(x^\rho) \neq 0]\right)^t \geq \frac{1}{2^{(dt)^{c_2}}}.$$

By setting $t = d^{\frac{c_2}{1-c_2}} \log^{\frac{1}{c_2-1}} d$, we have

$$\min_{x} \Pr_{\rho\sim\mathcal{D}^*}[f(x^\rho) \neq 0] \geq 2^{-(dt)^{c_2}/t} = 2^{-\log d} = 1/d. \qquad \blacktriangleleft$$

▶ **Lemma 24.** *Given positive integers t, K, and function $f : \{-1, 1\}^n \to \mathbb{R}$, let $\tilde{f}(x_1, \ldots, x_t) := \prod_{i=1}^t f(x_i)$. Then*

$$\max_{\tilde{\mathcal{D}}} \min_{\tilde{x}} \Pr_{\tilde{\rho}\sim\tilde{\mathcal{D}}}\left[\tilde{f}(\tilde{x}^{\tilde{\rho}}) \neq 0\right] = \left(\max_{\mathcal{D}} \min_{x} \Pr_{\rho\sim\mathcal{D}}[f(x^\rho) \neq 0]\right)^t$$

where $\mathcal{D}$ is a distribution of partial assignments $\rho \in \{-1, 1, \star\}^n$ with $|\rho| \leq K$, $\tilde{\mathcal{D}}$ is a distribution of partial assignments $\tilde{\rho} = (\rho_1, \ldots, \rho_t) \in (\{-1, 1, \star\}^n)^t$ with $|\rho_i| \leq K$ for all i.

Proof. By the min-max principle, we have

$$\max_{\mathcal{D}} \min_{x} \Pr_{\rho\sim\mathcal{D}}[f(x^\rho) \neq 0] = \min_{X} \max_{\rho} \Pr_{x\sim X}[f(x^\rho) \neq 0] := p.$$

Let $\mathcal{D}^*$ be the optimal distribution that reaches the maximum, X^* the optimal distribution that reaches the minimum. Let $\tilde{p} := \max_{\tilde{\mathcal{D}}} \min_{\tilde{x}} \Pr_{\tilde{\rho}\sim\tilde{\mathcal{D}}}\left[\tilde{f}(\tilde{x}^{\tilde{\rho}}) \neq 0\right]$.

Claim 1: $\tilde{p} \leq p^t$. By min-max principle,

$$\begin{aligned}\max_{\tilde{\mathcal{D}}} \min_{\tilde{x}} \Pr_{\tilde{\rho}\sim\tilde{\mathcal{D}}}\left[\tilde{f}(\tilde{x}^{\tilde{\rho}}) \neq 0\right] &= \min_{\tilde{X}} \max_{\tilde{\rho}} \Pr_{\tilde{x}\sim\tilde{X}}\left[\tilde{f}(\tilde{x}^{\tilde{\rho}}) \neq 0\right] \\ &= \min_{\tilde{X}} \max_{\rho_1,\ldots,\rho_t} \Pr_{(x_1,\ldots,x_t)\sim\tilde{X}}\left[f(x_1^{\rho_1}) \neq 0, \ldots, f(x_t^{\rho_t}) \neq 0\right].\end{aligned}$$

By setting $\tilde{X}$ to be $X^* \times X^* \times \cdots \times X^*$, we have

$$\begin{aligned}\max_{\tilde{\mathcal{D}}} \min_{\tilde{x}} \Pr_{\tilde{\rho}\sim\tilde{\mathcal{D}}}\left[\tilde{f}(\tilde{x}^{\tilde{\rho}}) \neq 0\right] &\leq \max_{\rho_1,\ldots,\rho_t} \Pr_{x_i\sim X^*}[f(x_1^{\rho_1}) \neq 0, \ldots, f(x_t^{\rho_t}) \neq 0] \\ &= \max_{\rho_1,\ldots,\rho_t} \prod_i \Pr_{x_i\sim X^*}[f(x_i^{\rho_i}) \neq 0] = \prod_t \max_{\rho_i} \Pr_{x_i\sim X^*}[f(x_i^{\rho_i}) \neq 0] = p^t.\end{aligned}$$

Claim 2: $\tilde{p} \geq p^t$. By setting $\tilde{\mathcal{D}}$ to be $\mathcal{D}^* \times \mathcal{D}^* \times \cdots \times \mathcal{D}^*$, we have

$$\begin{aligned}\max_{\tilde{\mathcal{D}}} \min_{\tilde{x}} \Pr_{\tilde{\rho}\sim\tilde{\mathcal{D}}}\left[\tilde{f}(\tilde{x}^{\tilde{\rho}}) \neq 0\right] &\geq \min_{x_1,\ldots,x_t} \Pr_{\rho_i\sim\mathcal{D}^*}[f(x_1^{\rho_1}) \neq 0, \ldots, f(x_t^{\rho_t}) \neq 0] \\ &= \min_{x_1,\ldots,x_t} \prod_i \Pr_{\rho_i\sim\mathcal{D}^*}[f(x_i^{\rho_i}) \neq 0] = \prod_i \min_{x_i} \Pr_{\rho_i\sim\mathcal{D}^*}[f(x_i^{\rho_i}) \neq 0] = p^t.\end{aligned}$$

Thus $\tilde{p} = p^t$ and $\tilde{p}$ can be reached by i.i.d. distribution $\mathcal{D}^* \times \cdots \times \mathcal{D}^*$. $\blacktriangleleft$

▶ Remark 25. By the same technique, the non-zero probability in Conjecture 1 can be further improved from $1/\mathrm{poly}(d)$ to $1 - 1/\mathrm{poly}(d)$, while still keeping $|\rho| \leq \mathrm{poly}(d)$.

4.3 Special cases of boolean functions

We also affirm Conjecture 1 for some special classes of functions.

▶ **Lemma 26.** *Conjecture 1 holds when f satisfies one of the following conditions:*

(a) *f can be expressed as the acceptance probability of a classical randomized decision tree with depth $\leq \mathrm{poly}(d)$ (Lemma 5).*

(b) *f is Boolean-valued, more generally when f takes at most $\mathrm{poly}(d)$ values (Lemma 11).*

(c) *f is a symmetric function.*

Proof.

Part (a). Recall that a randomized decision tree can be viewed as a distribution μ over deterministic decision trees, such that the tree is evaluated by (i) first sampling a deterministic decision tree from μ, and (ii) then evaluating this deterministic decision tree. Since $f(x)$ is the probability that the output of the evaluation is "yes", and f is non-zero, there exists a deterministic decision tree T with $\mu(T) > 0$ such that at least one of its leaves is labeled with "yes". We can set ρ to be the partial assignment corresponding to the path from the root to the leaf. Then $f(x^\rho) \neq 0$ for any x.

Part (b). Without loss of generality, assume $\|f\|_\infty = 1$. Since f takes at most $\mathrm{poly}(d)$ values, there exists a gap $b - a \geq 1/\mathrm{poly}(d)$ such that $f(x) \in [-1, a] \cup [b, 1]$ for all x. Define $f'(x) := \left(1 + \left|\frac{a+b}{2}\right|\right)^{-1} \left(f(x) - \frac{a+b}{2}\right)$. Then $f'(x) \in [-1, -1/\mathrm{poly}(d)] \cup [1/\mathrm{poly}(d), 1]$ for all x. Define a boolean function

$$\tilde{f}(x) := \begin{cases} -1 & \text{if } f'(x) < 0 \\ 1 & \text{if } f'(x) > 0 \end{cases}.$$

Then $\tilde{f}$ is $(1 - 1/\mathrm{poly}(d))$-approximated by f'. f' can be amplified to a degree-$\mathrm{poly}(d)$ polynomial that 1/3-approximates $\tilde{f}$. Thus we have $\widetilde{\deg}(\tilde{f}) \leq \mathrm{poly}(d)$, which further implies $\tilde{f}$ can be computed by a decision tree with depth $\mathrm{poly}(d)$ [11].

(i) If $a < 0 < b$, then $f(x) \neq 0$ alway holds. (ii) If $a \geq 0$, by Part (a), there exists a partial assignment $|\rho| \leq \mathrm{poly}(d)$ such that $\tilde{f}(x) = 1$, which implies $f(x^\rho) \geq b > 0$ for any x. (iii) If $b \leq 0$, by Part (a), there exists a partial assignment $|\rho| \leq \mathrm{poly}(d)$ such that $\tilde{f}(x) = -1$, which implies $f(x^\rho) \leq a < 0$ for any x.

Part (c). Let $x_S = x_{i_1} x_{i_2} \cdots x_{i_d}$ be a maximal monomial of f, and ρ_k be the partial assignment such that $\rho_k(i_1) = \cdots = \rho_k(i_k) = 1$ and $\rho_k(i_{k+1}) = \cdots = \rho_k(i_d) = -1$. By Lemma 22, for any x, there exists a ρ with $\mathsf{supp}(\rho) = S$ such that $f(x^\rho) \neq 0$. Since f is symmetric, we have $f(x^{\rho_j}) = f(x^\rho) \neq 0$ where j is the number of 1's in ρ. Let $\mathcal{D}$ be the uniform distribution over $\{\rho_0, \rho_1, \ldots, \rho_d\}$. Then $\Pr_{\rho\sim\mathcal{D}}[f(x^\rho) \neq 0] \geq 1/(d+1)$ for any x. ◀

The above lemma implies the non-existence of perfect complete QPKE in QROM which the key generation takes special forms, e.g., the special cases 1 and 3 mentioned in the introduction.

▶ **Corollary 27.** *Perfect complete QPKE does not exist in the QROM, if one of the following holds:*

(1) Gen *only makes classical queries. (Corollary 6)*

(2) Gen *is uniform, i.e., it satisfies (i)* $|\mathsf{supp}(\mathsf{Gen}^H)| = |\mathsf{supp}(\mathsf{Gen}^{H'})|$ *for every pair of H, H', and (ii)* Gen^H *is uniform over* $\mathsf{supp}(\mathsf{Gen}^H)$ *with a non-zero probability. (Corollary 12)*

Proof. Consider the polynomial $g(H)$ defined in the proof of Theorem 17, which is defined to be the probability of $\mathsf{Gen}^{H^{\rho_E}}$ outputting (sk', m_0).

(1) g can be expressed as the acceptance probability of a classical randomized decision tree with depth d. Then the proposition follows from Part (a) of Lemma 26.
(2) g takes only two values 0 and $1/|\mathsf{supp}(\mathsf{Gen}^H)|$. Then the proposition follows from Part (b) of Lemma 26. ◀

4.4 More evidences for Conjecture 13

Additionally, we provide evidence for the equivalent form Conjecture 13 (and consequently for Conjecture 1). The first piece of evidence is from the anti-concentration result for low-degree polynomials by Meka, Nguyen and Vu [20], which implies Conjecture 13 when X is uniform distribution on $\{-1,1\}^n$, except that $|\rho|$ is allowed to be as large as d^{8d+4} instead of $|\rho| \leq \mathrm{poly}(d)$.

▶ **Theorem 28** ([20]). *Let $f : \{-1,1\}^n \to \mathbb{R}$ be a nonconstant polynomial with degree d, and U denote the uniform distribution on $\{-1,1\}^n$. If $\mathrm{rank}(f) \geq d^{8d+2}$, then $\Pr_{x\sim U}[f(x) \neq 0] = 1 - o(1)$.*

▶ **Lemma 29.** *For any nonconstant function $f : \{-1,1\}^n \to \mathbb{R}$ with degree d, there exists a partial assignment with $|\rho| \leq d^{8d+4}$ such that $\Pr_{x\sim U}[f(x^\rho) \neq 0] \geq 1 - o(1)$.*

Proof. We construct a partial assignment ρ by the following algorithm, which contains at most d rounds and each round reduces $\deg(f)$ by at least 1. Denote the function at round t by f^t. Initia,lly ρ is empty. At round t,

1. If $\mathrm{rank}(f^t) \geq d^{8d+2}$, the algorithm stops. By Theorem 28, we have
$$\Pr_{x\sim U}[f(x^\rho) \neq 0] = \Pr_{x\sim U}[f^t(x) \neq 0] = 1 - o(1).$$
2. If $\mathrm{rank}(f^t) < d^{8d+2}$, let T be the variables contained a maximum disjoint set of maximal monomials of f^t. The algorithm assigns all the variables in T such that the function after the assignment is still nonconstant and adds them to ρ. This step increases $|\rho|$ by $|T| = \deg(f^t)\mathrm{rank}(f^t) < d^{8d+3}$. For any maximal monomial S of f^t, we have $S \cap T \neq \emptyset$ because otherwise $S \cup T$ is a disjoint set of maximal monomials larger than T. Thus this step reduces $\deg(f^t)$ by at least one.

Because there are at most d rounds, we have $|\rho| \leq d^{8d+4}$. ◀

The second evidence is that Conjecture 13 holds for any function f with large variance when X is uniform.

▶ **Lemma 30.** *For any nonconstant function $f : \{-1,1\}^n \to \mathbb{R}$ with degree d and $\mathsf{Var}(f) \geq \|f\|_\infty^2/\mathrm{poly}(d)$, there exists a partial assignment with $|\rho| \leq \mathrm{poly}(d)$ such that $\Pr_{x\sim U}[f(x^\rho) \neq 0] \geq 1/\mathrm{poly}(d)$.*

Proof. $\mathsf{Var}(f) = \mathbb{E}_x[f^2(x)] - (\mathbb{E}_x f(x))^2 \leq \mathbb{E}_x[f^2(x)] \leq \Pr_x[f(x) \neq 0]\|f\|_\infty^2$. Thus $\Pr_x[f(x) \neq 0] \geq \frac{\mathsf{Var}(f)}{\|f\|_\infty^2} \geq 1/\mathrm{poly}(d)$. ◀

References

1 Scott Aaronson and Andris Ambainis. Forrelation: A problem that optimally separates quantum from classical computing. *SIAM J. Comput.*, 47(3):982–1038, 2018. doi:10.1137/15M1050902.
2 Noga Alon. Combinatorial nullstellensatz. *Comb. Probab. Comput.*, 8(1–2):7–29, January 1999.

3 Prabhanjan Ananth, Luowen Qian, and Henry Yuen. Cryptography from pseudorandom quantum states. In *Annual International Cryptology Conference*, pages 208–236. Springer, 2022. doi:10.1007/978-3-031-15802-5_8.

4 Srinivasan Arunachalam, Jop Briët, and Carlos Palazuelos. Quantum query algorithms are completely bounded forms. *SIAM J. Comput.*, 48(3):903–925, 2019. doi:10.1137/18M117563X.

5 Per Austrin, Hao Chung, Kai-Min Chung, Shiuan Fu, Yao-Ting Lin, and Mohammad Mahmoody. On the impossibility of key agreements from quantum random oracles. In *Annual International Cryptology Conference*, pages 165–194. Springer, 2022. doi:10.1007/978-3-031-15979-4_6.

6 Khashayar Barooti, Alex B. Grilo, Loïs Huguenin-Dumittan, Giulio Malavolta, Or Sattath, Quoc-Huy Vu, and Michael Walter. Public-key encryption with quantum keys, 2023. doi:10.48550/arXiv.2306.07698.

7 James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. On the round complexity of secure quantum computation. In *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part I 41*, pages 406–435. Springer, 2021. doi:10.1007/978-3-030-84242-0_15.

8 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001. doi:10.1145/502090.502097.

9 Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997. doi:10.1137/S0097539796300933.

10 Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science*, 560:7–11, December 2014. doi:10.1016/j.tcs.2014.05.025.

11 Harry Buhrman and Ronald De Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002. doi:10.1016/S0304-3975(01)00144-X.

12 Andrea Coladangelo. Quantum trapdoor functions from classical one-way functions. *arXiv preprint arXiv:2302.12821*, 2023. URL: https://eprint.iacr.org/2023/282.

13 Whitfield Diffie and Martin E Hellman. New directions in cryptography. In *Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman*, pages 365–390. 2022. doi:10.1145/3549993.3550007.

14 Omar Fawzi and Renato Renner. Quantum conditional mutual information and approximate markov chains. *Communications in Mathematical Physics*, 340(2):575–611, 2015.

15 Alex B Grilo, Huijia Lin, Fang Song, and Vinod Vaikuntanathan. Oblivious transfer is in miniqcrypt. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 531–561. Springer, 2021. doi:10.1007/978-3-030-77886-6_18.

16 Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 44–61, 1989. doi:10.1145/73007.73012.

17 Fuyuki Kitagawa, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa. Quantum public-key encryption with tamper-resilient public keys from one-way functions. *arXiv preprint arXiv:2304.01800*, 2023. URL: https://eprint.iacr.org/2023/490.

18 Longcheng Li, Qian Li, Xingjian Li, and Qipeng Liu. How (not) to build qpke in minicrypt. In *Annual International Cryptology Conference*. Springer, 2024. doi:10.1007/978-3-031-68394-7_6.

19 Giulio Malavolta and Michael Walter. Robust quantum public-key encryption with applications to quantum key distribution. Cryptology ePrint Archive, Paper 2023/500, 2023. doi:10.1007/978-3-031-68394-7_5.

20 Raghu Meka, Oanh Nguyen, and Van Vu. Anti-concentration for polynomials of independent random variables. *Theory of Computing*, 12:11, 2016. doi:10.4086/toc.2016.v012a011.

21 Gatis Midrijanis. Exact quantum query complexity for total boolean functions, 2004. arXiv:quant-ph/0403168.

22 Tomoyuki Morimae and Takashi Yamakawa. One-wayness in quantum cryptography. *arXiv preprint arXiv:2210.03394*, 2022. URL: https://eprint.iacr.org/2022/1336.

23 Tomoyuki Morimae and Takashi Yamakawa. Quantum commitments and signatures without one-way functions. In *Annual International Cryptology Conference*, pages 269–295. Springer, 2022. doi:10.1007/978-3-031-15802-5_10.

Information Design with Unknown Prior

Tao Lin
John A. Paulson School of Engineering and Applied Sciences,
Harvard University, Cambridge, MA, USA

Ce Li
Department of Economics, Boston University, MA, USA

Abstract

Classical information design models (e.g., Bayesian persuasion and cheap talk) require players to have perfect knowledge of the prior distribution of the state of the world. Our paper studies repeated persuasion problems in which the information designer does not know the prior. The information designer learns to design signaling schemes from repeated interactions with the receiver. We design learning algorithms for the information designer to achieve no regret compared to using the optimal signaling scheme with known prior, under two models of the receiver's decision-making:

(1) The first model assumes that the receiver knows the prior and can perform posterior update and best respond to signals. In this model, we design a learning algorithm for the information designer to achieve $O(\log T)$ regret in the general case, and another algorithm with $\Theta(\log\log T)$ regret in the case where the receiver has only two actions. Our algorithms are based on multi-dimensional and conservative binary search techniques, which circumvent the $\Omega(\sqrt{T})$ limitation of empirical estimation in previous works.

(2) The second model assumes that the receiver does not know the prior either and employs a no-regret learning algorithm to take actions. Bayesian persuasion and cheap talk are equivalent under this no-regret learning receiver model. We show that the information designer can achieve regret $O(\sqrt{\mathrm{rReg}(T)T})$, where $\mathrm{rReg}(T) = o(T)$ is an upper bound on the receiver's learning regret. The algorithm is based on exploration + robustification. The $O(\sqrt{\mathrm{rReg}(T)T})$ regret bound is tight even when the information designer knows the prior [1].

Our work thus provides a learning foundation for the problem of information design with unknown prior.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory

Keywords and phrases information design, Bayesian persuasion, online learning, unknown prior

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.72

Category Extended Abstract

Related Version *Full Version*: `https://arxiv.org/pdf/2410.05533` [2]

References

1 Tao Lin and Yiling Chen. Generalized Principal-Agent Problem with a Learning Agent, 2024. `doi:10.48550/arXiv.2402.09721`.

2 Tao Lin and Ce Li. Information design with unknown prior, 2024. `arXiv:2410.05533`, `doi:10.48550/arXiv.2410.05533`.



© Tao Lin and Ce Li;
licensed under Creative Commons License CC-BY 4.0



On White-Box Learning and Public-Key Encryption

Yanyi Liu
Cornell Tech, New York, NY, USA

Noam Mazor
Tel Aviv University, Israel

Rafael Pass
Cornell Tech, New York, NY, USA
Technion, Haifa, Israel
Tel Aviv University, Israel

Abstract

We consider a generalization of the Learning With Error problem, referred to as the *white-box learning problem*: You are given the code of a sampler that with high probability produces samples of the form $y, f(y) + \epsilon$ where ϵ is small, and f is computable in polynomial-size, and the computational task consist of outputting a polynomial-size circuit C that with probability, say, $1/3$ over a new sample y' according to the same distributions, approximates $f(y')$ (i.e., $|C(y') - f(y')|$ is small). This problem can be thought of as a generalizing of the Learning with Error Problem (LWE) from linear functions f to polynomial-size computable functions.

We demonstrate that worst-case hardness of the white-box learning problem, conditioned on the instances satisfying a notion of computational shallowness (a concept from the study of Kolmogorov complexity) not only *suffices* to get public-key encryption, but is also *necessary*; as such, this yields the first problem whose worst-case hardness characterizes the existence of public-key encryption. Additionally, our results highlights to what extent LWE "overshoots" the task of public-key encryption.

We complement these results by noting that worst-case hardness of the same problem, but restricting the learner to only get black-box access to the sampler, characterizes one-way functions.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases Public-Key Encryption, White-Box Learning

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.73

Related Version *Full Version*: `https://eprint.iacr.org/2024/1931` [24]

Funding *Yanyi Liu*: Research partly supported by NSF CNS-2149305.
Noam Mazor: Research partly supported by NSF CNS-2149305, AFOSR Award FA9550-23-1-0312 and AFOSR Award FA9550-23-1-0387 and ISF Award 2338/23.
Rafael Pass: Supported in part by NSF Award CNS 2149305, AFOSR Award FA9550-23-1-0387, AFOSR Award FA9550-23-1-0312 and AFOSR Award FA9550-24-1-0267 and ISF Award 2338/23. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government, or the AFOSR.

1 Introduction

Public-Key Encryption (PKE) [12, 31] is one of the most central cryptographic primitives enabling secure communication on the Internet: it is the primitive that enables entities that have not physically met to engage in confidential conversations and collaborations.

In contrast to private-key primitives, such as symmetric-key encryption and digital signatures, that can be securely build from the minimal primitive of one-way functions (and for which many candidate problems are known), we only know of a handful of candidate


© Yanyi Liu, Noam Mazor, and Rafael Pass;
licensed under Creative Commons License CC-BY 4.0



hard problems from which public-key encryption can be constructed. More specifically, these include (a) *number-theory problems* based on either factoring [31, 29] or discrete logarithms [12, 13], (b) coding-theory based problems [27], (c) lattice problems as finding shortest/longest vectors in lattices [1, 30, 8], and (d) noisy linear-algebra based problems [2, 5]. Out of these, the number-theory based problems can be efficiently solved by quantum algorithms [32], and the coding-theory, lattice and noisy linear algebra problems are all very related – in essence, they can all be viewed as different instances of solving noisy systems of *linear* equations (on either particular natural distributions, or even in the worst-case, when restricting attention to systems of equations satisfying the condition that appropriate solutions exist.[1]

The main purpose of this paper is to provide an assumption that generalizes all these assumptions (i.e., is implied by all of them), yet suffices for obtaining secure PKE. Indeed, the main result of this paper is that hardness of a notion of *white-box learning* achieves this goal.

White-box Learning

Perhaps the most common noisy linear algebra-based assumption is the hardness of the *Learning With Error (LWE)* problem [30], which, in essence, stipulates the hardness of recovering a vector $\mathbf{x}$ given $\mathbf{A}, \mathbf{A}\mathbf{x} + \mathbf{e}$, where $\mathbf{A}$ is a matrix, $\mathbf{e}$ is some "small" noise vector and all arithmetic is modulo some prime p. In more detail, we typically require an stronger condition: not only that is hard to recover x, but also that it is hard to compute a value "close" to $\mathbf{a}^T\mathbf{x}$ for a random vector $\mathbf{a}$. In other words, we can think of $\mathbf{x}$ as the description of a function $f_{\mathbf{x}}(\mathbf{a}) = \mathbf{a}^T\mathbf{x}$ that we are trying to *improperly* approximately learn given noisy samples – thus the name "learning with error".

In fact, there is also a different way to think about the LWE problem that will be useful for our purposes (which follows from the construction of Regev's PKE [30]): We are given the code of a sampler P_x that enables providing samples of the form $(\mathbf{a}, f_{\mathbf{x}}(\mathbf{a}) + \mathbf{e})$, and the goal is to approximate $f_{\mathbf{x}}(\mathbf{a}')$ on a new fresh sample $\mathbf{a}'$ according to the same distribution as $\mathbf{a}$.[2] We refer to this alternative way of thinking of LWE as an instance of *(improper) white-box learning*, where, more generally, we are given the code of a sampler P that generates noisy samples of the form $(y, f(y) + \epsilon)$ and the goal is to approximate $f(y')$ for a fresh sample y' according to the same distribution, using a polynomial-size circuit. In essence, this problem is generalizing the LWE problem from (improperly) learning *linear functions* from noisy samples, to (improperly) *learning a polynomial-size circuit* from noisy samples, and given white-box access to the sampler[3]; the white-box access feature can be viewed as a generalization of Valiant classic PAC-learning model [35] to a setting where the learner not only gets random samples, but also gets the code of the sampler.

In more detail, given a sampler circuit P that samples "labeled instances" (y, z) (where we think of $z \in \mathbb{N}$ as a, perhaps, noisy label for y), let $\mathsf{Comp}_\epsilon^\Delta(P)$ denote the set of circuits f that with probability $1 - \epsilon$ over $(y, z) \leftarrow P$ satisfy the property that $|z - f(y)| \leq \Delta$ (when interpreting both z and $f(z)$ as elements in $\mathbb{N}$). For a function $\Delta\colon \{1\}^* \to \mathbb{N}$, let Δ-$\mathsf{WBLearn}$ denote the following learning problem:

---

[1] In more detail, we require worst-case hardness to hold when conditioning on instances that define a lattice where the shortest vector is long compared to amount of noise.

[2] The reason for this is that given $\mathbf{A}, \mathbf{A}\mathbf{x} + \mathbf{e}$, we can generate noisy samples of $f_{\mathbf{x}}(\mathbf{a}')$ by taking linear combinations. See [30] and the full version of this paper for more details.

[3] As we will discuss in more detail later, it is also more general than the LWE problem in the aspect the LWE *sampler* has a particular form, but we here may consider more general classes of samplers.

- **Input:** Circuit $P \in \{0,1\}^n$, with the promise that there exists a circuit $\widehat{C} \in \{0,1\}^n$ such that $\widehat{C} \in \mathsf{Comp}^{\Delta(1^n)/n^{10}}_{n^{-10}}(P)$.
- **Output:** Circuit $C \in \mathsf{Comp}^{\Delta(1^n)}_{1/3}(P)$.[4]

In other words, we are given a sampler that with very high $(1 - n^{-10})$ probability outputs labelled samples where the label is very close $(\Delta(1^n)/n^{10})$ to being correct, and the goal is to, given the code of the sampler P, simply find a circuit C that with probability $2/3$ approximates the label within $\Delta(1^n)$ (i.e., with a factor n^{10} higher error). We use $\mathsf{ExactWBLearn}$ to denote Δ-$\mathsf{WBLearn}$ when $\Delta(1^n) = 0$.

Hardness of Learning v.s. (Public-Key) Cryptography

Roughly speaking, the main result of this paper will be to show that under certain restrictions on samplers P (which come from the study of *time-bounded Kolmogorov complexity* [23, 34, 11, 22, 17, 33] and in particular the notion of *computational depth* [4]), this generalization of the LWE problem not only suffices to realize a PKE, but will also be necessary. In more detail, worst-case hardness of Δ-$\mathsf{WBLearn}$ under these conditions will *characterize* the existence of public-key encryption. As such, our results yield insight on the extent with which LWE "overshoots" the task of public-key encryption.

We highlight that we are not the first ones to consider connections between learning-theory and cryptography; however, as far as we know, all earlier connections were between the hardness of learning and *private-key cryptography* (i.e., the notion of one-way functions), as opposed to public-key cryptography. Indeed, classic results from [21, 7] demonstrate the equivalence of the hardness of a notion of *average-case* PAC-learning of polynomial-size circuits (i.e., black-box learning) and one-way functions.[5] In contrast, our focus here is on PKE, and instead of considering black-box learning, we consider white-box learning. An additional difference is that we consider worst-case hardness, as opposed to average-case hardness, of the learning theory problem. As was recently shown in [18], this issue can be overcome (in the context of characterizing one-way functions) through the use of a (alternative) notion of computational depth (more details on the relationship with this work below). To put out result in context, we additionaly show that exactly the same problem that we demonstrate characterizes PKE, also characterizes one-way functions once modified to only provide the learner black-box access to the sampler.

1.1 Our Results

Towards explaining our results in more detail, let us first recall the notion of *computational depth* [4]. Let the *t-computational depth* of x, denoted $cd^t(x)$, be defined as $cd^t(x) = \mathrm{K}^t(x) - \mathrm{K}(x)$ where $\mathrm{K}(x)$ denotes the Kolmogorov complexity of x and K^t denotes the t-bounded Kolmogorov complexity of x. That is, $cd^t(x)$ measures how much more x could be compressed if the decompression algorithm may be computationally unbounded, as opposed to it running in time bounded by $t(|x|)$.

We will focus on instances P of the Δ-$\mathsf{WBLearn}$ problem having the property that $(P, \widehat{C})$ is "computationally shallow", where $\widehat{C}$ is a circuit that agrees with P with high probability: let $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t}$ denote Δ-$\mathsf{WBLearn}$ with the additional promise that $cd^t(P, \widehat{C}) \leq 2 \log n$.[6]

[4] We remark that for efficient algorithms, outputting a circuit that approximates a function f is essentially equivalent to approximating the value of $f(y')$ on a random sample y'. Indeed, the circuit implementation of an algorithm for the latter task, is a valid output for the first task.

[5] And the results of [20] can be thought of as characterizing one-way functions through a different type of learning.

[6] We note that there is nothing special about the constant 2; it can be anything that is strictly larger

Characterizing PKE through Hardness of White-Box Learning

Our main results is that the hardness of this problem *characterizes* the existence of public-key encryption:

▶ **Theorem 1.** *Let $\epsilon > 0$ be a constant and let $\Delta\colon \{1\}^* \to \mathbb{N}$ be an efficiently computable function such that $\Delta(1^n) \leq 2^{n^{(1-\epsilon)}}$, and $t\colon \mathbb{N} \to \mathbb{N}$ be polynomial such that $t(n) \geq n^{1+\epsilon}$. Then, following are equivalent:*

- *PKE exists.*
- Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$.

Computational depth is typically thought of as measure of "unnaturality" of strings: strings with low computational depth are considered "natural" and those with high computational depth are considered "unnatural".[7] Given this interpretation, our characterization of public key encryption is thus in terms of the worst-case hardness of white-box learning for "natural" instances.

As far as we know, this thus yields the first problem whose worst-case hardness not only suffices for public-key encryption (such as e.g., [30]) but also is *necessary*.

On the Use of Computational Depth

We note that Antunes and Fortnow [3] elegantly used computational depth to connect worst-case hardness of a problem when restricting attention to elements with small computational depth and *errorless* average-case hardness on sampleable distributions; errorless hardness, however, is not sufficient for cryptographic applications. Nevertheless, inspired by the work of [3], worst-case hardness conditioned on instances with small computational depth was used in [26] (and independently using a variant of this notion in [18]) to characterize one-way functions; additionally, an (interactive) variant of such a notion was also implicitly used in [6] to characterize key exchange protocols. Our techniques are similar to those employed in [26, 6] but instead of applying them to study the hardness of a time-bounded Kolmogorov complexity problem (following [25]), we here instead apply them to study a learning theory problem (namely, white-box learning).

We note that learning theory problems conditioned on small computational depth were recently used in [18] to characterize one-way functions, but our techniques here are more similar to [26, 6]. In particular, [18] does not actually use the standard notion of computational depth but instead define a new alternative variant; in contrast, we here rely on just the standard notion.

Relating Exact and Approximate White-Box Learning

Note that in Theorem 1, the equivalence hold for any (sufficiently small) choice of Δ, as such we directly get as a corollary the equivalence of Exact and Approximate White-box Learning:

---

than 1.

[7] The reason why low computational depth captures "natural string" is as follows: random strings are known to have low computational depth; furthermore, known results (c.f. slow growth laws [18]) show (at least under derandomization assumptions) that one needs to have a long running time to even find a string with high computational depth. So strings with high computational depth are rare and "hard to find", which is why they can be thought of as "unnatural".

▶ **Corollary 2.** *Let $\epsilon > 0$ be a constant and let $\Delta\colon \{1\}^* \to \mathbb{N}$ be any efficiently computable function such that $\Delta(1^n) \leq 2^{n^{(1-\epsilon)}}$, and let $t\colon \mathbb{N} \to \mathbb{N}$ be any polynomial such that $t(n) \geq n^{1+\epsilon}$. Then the following are equivalent:*

- $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$
- $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$

Bounded-Degree Learning and LWE

While in the Δ-WBLearn problem, we allow the function we are trying to learn to be any polynomial-size circuit, we may also consider a restricted version of the problem, denoted $\Delta\text{-}\mathsf{WBLearn}^d_q$, where we restrict attention to functions that can be computed by a degree d polynomial, and we assume that arithmetic is now over $\mathbb{Z}_q$.

We first remark that our main theorem can next be generalized (basically using padding) to show that it suffices to use learning of degree-n^ϵ polynomials to characterize PKE:

▶ **Theorem 3.** *Let $\epsilon > 0$ be a constant and let $\Delta, q\colon \{1\}^* \to \mathbb{N}$ be efficiently computable functions with $\Delta(1^n) \leq q(1^n)/4$ and $q(1^n) \leq 2^{n^{(1-\epsilon)}}$, and $t\colon \mathbb{N} \to \mathbb{N}$ be polynomial such that $t(n) \geq n^{1+\epsilon}$. The following are equivalent:*

- *PKE exists.*
- $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$.
- $\Delta\text{-}\mathsf{WBLearn}^n_q|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$.
- $\Delta\text{-}\mathsf{WBLearn}^{n^\epsilon}_q|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$.

Additionally, as informally discussed above, we note that the hardness of the LWE problem, implies hardness of the $\Delta\text{-}\mathsf{WBLearn}^1_q$ problem (i.e., white-box learning of *linear* functions.)

▶ **Lemma 4** (Informal)**.** *Assuming the hardness of LWE, there exists a polynomial t such that $\Delta\text{-}\mathsf{WBLearn}^1_q|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$ where $\Delta = q/4$.*

Lemma 4 thus shows that white-box learning of linear functions is at least as weak an assumption as LWE. At first sight, one would expect that a converse result may also hold due to known worst-case to average-case reductions for the LWE problem [30, 28, 9] and thus that LWE is equivalent to white-box learning of linear functions. However, the problem with proving the converse direction is that the known worst-case to average-case reductions for LWE only work when the LWE instance defines a lattice where the shortest vector is long compared to amount of noise. Instances sampled from P may not necessarily satisfy this promise, and thus is it not clear how to use an LWE oracle to generally solving white-box learning of linear functions. Thus, it would seem that hardness even of just $\Delta\text{-}\mathsf{WBLearn}^1_q|_{\mathsf{CS}^t}$ is seemingly a weaker (and therefore more general) assumption than LWE. (Of course, it may be that a stronger worst-case to average-case reduction can be established for the LWE problem, in which case equivalence would hold.)

We finally investigate what happens in the regime of "intermediate-degree" polynomials. We remark that using standard linearization techniques, the constant-degree problem is equivalent to the case of degree 1:

▶ **Lemma 5.** *For every constant $d \in \mathbb{N}$, and functions $t, q, \Delta\colon \mathbb{N} \to \mathbb{N}$, there exist $t', q', \Delta'\colon \mathbb{N} \to \mathbb{N}$ such that*

$$\Delta\text{-}\mathsf{WBLearn}^d_q|_{\mathsf{CS}^t} \leq_p \Delta'\text{-}\mathsf{WBLearn}^1_{q'}|_{\mathsf{CS}^{t'}},$$

and

$$\Delta\text{-}\mathsf{WBLearn}^d_q \leq_p \Delta'\text{-}\mathsf{WBLearn}^1_{q'}.$$

Black-box Learning

Finally, to put our results in context, we consider the standard PAC learning model [35] where the learner only get access to samples: Let Δ-BBLearn denote identically the same problem as WBLearn with the exception that the learner gets oracle access (i.e., black-box access) to the sampler (as opposed to white-box access to the sampler). This notion is equivalent to the notion of improper Δ-approximate PAC learning for polynomial-size circuits (and when $\Delta = 0$ to simply improper PAC learning).

As before, let $\Delta\text{-BBLearn}|_{\mathsf{CS}^t}$ denote the problem Δ-BBLearn with the additional promise that $cd^t(P, \widehat{C}) \leq 2 \log n$, and let ExactBBLearn and $\text{ExactBBLearn}|_{\mathsf{CS}^t}$ to denote Δ-BBLearn and $\Delta\text{-BBLearn}|_{\mathsf{CS}^t}$ when $\Delta(1^n) = 0$.

The following theorem can be viewed as the worst-case analog of the classic result of [21, 7] characterizing one-way functions through the hardness of *average-case PAC learning*.

▶ **Theorem 6.** *Let $\epsilon > 0$ be a constant and let $\Delta\colon \{1\}^* \to \mathbb{N}$ be any efficiently computable function such that $\Delta(1^n) \leq 2^{n^{(1-\epsilon)}}$, and let $t\colon \mathbb{N} \to \mathbb{N}$ be any polynomial such that $t(n) \geq n^{1+\epsilon}$. Then the following are equivalent:*

- *One-way function exists*
- $\Delta\text{-BBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$

As mentioned, [18] also recently obtained a worst-case characterization of one-way functions through a learning problem, and using a notion of computational depth. The problems, however, are somewhat different. As opposed to [18], we here consider the standard PAC learning problem (whereas they consider a more general learning problem), and condition on the standard notion of low computational depth (whereas they condition on a new notion of low computational depth that they introduce).[8]

We remark that our Theorem 6 differs from [21, 7] not only in the worst-case condition, but also generalizes those results in the sense that we handle the hardness of Δ-approximate learning for any Δ. As a consequence, we again get an equivalence of approximate and exact black-box learning:

▶ **Corollary 7.** *Let $\epsilon > 0$ be a constant and let $\Delta\colon \{1\}^* \to \mathbb{N}$ be any efficiently computable function such that $\Delta(1^n) \leq 2^{n^{(1-\epsilon)}}$, and let $t\colon \mathbb{N} \to \mathbb{N}$ be any polynomial such that $t(n) \geq n^{1+\epsilon}$. Then the following are equivalent:*

- $\text{ExactBBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$
- $\Delta\text{-BBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$

Open Problems

We leave as an intriguing open problem the question of whether white-box learning of polynomial, or even logarithmic-degree polynomials, also can be collapsed down to the case of constant-degree functions (and thus to linear functions); if this were possible, it would show that PKE, in essence, inherently requires the structure of the LWE problem.

Additionally, as discussed above, even when just restricting attention to learning linear functions, our learning problem generalizes LWE. It would appear that despite this, cryptographic applications of LWE (e.g., to obtain fully homomorphic encryption [14, 10]) nevertheless may still be possible from this generalized version; we leave an exploration of this for future work.

[8] Of course, on a conceptual level, these results are similar; the key point we are trying to make here is that exactly the same learning problem characterizes either one-way functions or PKE, depending on whether the learner gets black-box or white-box access to the sampler.

Finally, it is an intriguing open problem to relate black-box and white-box learning. By our results, doing so is equivalent to relating public-key and secret-key encryption. We note that relating black-box and white-box learning is interesting even just for the case of linear functions (which by our results is equivalent to $O(1)$-degree polynomials). Indeed, Regev's construction of a PKE [30] can be thought of a reduction from black-box learning to white-box learning of linear functions for a *specific* distribution; it is possible that a similar reduction may be applicable more generally.

1.2 Proof Overview

We here provide a detailed proof overview for the proof of Theorem 1. For simplicity, we will show the equivalence between PKE and the worst-case hardness of the exact version, $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$. We start with the construction of PKE based on the hardness of $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$.

Weak PKE

First, we use the well-known fact that a PKE is simply a two-rounds key-agreement protocol. Moreover, by the Key-agreement Amplification Theorem of Holenstein [19] and an application of the Goldreich-Levin theorem [15], to obtain (full-fledged) PKE, it suffices to obtain a weak form of two-rounds key-agreemnt, which we simply refer to as *Weak PKE* defined as follows: There exist some $\epsilon = 1/\text{poly}$ such that *agreement* between A and B happens with probability $1 - \epsilon$. *Security* requires that Eve cannot guess the key (output by Alice) with probability better than $1 - 20\epsilon$.

The Weak PKE protocol

We will next define the weak PKE protocol. We note that this construction resembles the universal key-agreement construction from [16], but with some crucial difference that enable our security proof. The parties A and B on input n perform the following steps:

- *Sample random program:* A samples a random length $\lambda \in [2n]$, and a random program Π of length λ.
- *Run random program:* Next, A runs the program Π for at most $t(n)$ steps to get an output, and interprets the output as a pair of circuits $P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell$ and $C\colon \{0,1\}^k \to \{0,1\}^\ell$. (Think of P as the sampler for a white-box learning instance, and of C as a potential solution to the problem.) If the output of Π is not a valid encoding of such a pair, A sets $P = P_0$ and $C = C_0$ for two fixed circuits P_0, C_0 that always output 0.
- *Agreement estimation:* A estimates the "agreement probability" of P and C (i.e., checking whether C indeed is a good solution): it samples n^{20} random inputs $w_1, \ldots, w_{n^{20}}$ for P, and computes $(x_i, s_i) = P(w_i)$. It then lets $\widehat{\alpha} = \Pr_{i \leftarrow [n^{20}]}[C(x_i) = s_i]$. If $\widehat{\alpha} \leq 1 - n^{-9}$, A reset the pair $(P, C) = (P_0, C_0)$.
- *First message:* A sends (the "sampler") P to B.
- *Second message:* B applies P on a random input w, and computes $(x, s) = P(w)$. It then sends x to A.
- *Outputs:* A outputs $C(x)$ and Bob outputs s.

Agreement

We claim that with probability $1 - 1/n^8$, Alice and Bob will agree (i.e., the final outputs are the same). Note that if $(P, C) = (P_0, C_0)$, Alice and Bob always agree. Moreover, let $\alpha = \Pr_{(x,s) \leftarrow P(U_r)}[C(x) = s]$. Then, the probability of Alice and Bob to agree given that

Alice uses (P, C) as the circuits in the protocol, is exactly α. We observe that by the Chernoff bound, the probability that $\widehat{\alpha}$ is far from α is small, and thus Alice uses (P, C) only when α is larger than $1 - n^{-8}$.

Security

We claim that Eve that can guess s with probability $1 - n^{-7}$ can be used to solve $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$. In more detail, consider such Eve, and let P be an input for $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$. The idea is to construct an algorithm Learner, that on input P outputs a circuit C, such that $C(x)$ simply simulates Eve on the messages P and x. C then outputs Eve's output. For simplicity, in the following we assume that Eve (and thus Learner) is deterministic. (We note that this is a *non-black-box* reduction: We are using the code of Eve to generate C – in particular, we are including the code of Eve into this circuit.[9])

Let $\mathbf{P}$ be a random variable distributed according to the same distribution of the first message in the above protocol. By assumption, we have that

$$\Pr[C' = \mathsf{Learner}(\mathbf{P}); (x, s) \leftarrow \mathbf{P}(U_r); C'(x) = s] \geq 1 - n^{-7}.$$

It follows by a simple averaging argument that

$$\Pr_{\mathbf{P}, C' = \mathsf{Learner}(\mathbf{P})}[\Pr_w[(x, s) \leftarrow \mathbf{P}(w); C'(x) = s] \geq 2/3] \geq 1 - 3n^{-7}.$$

Namely, Learner solves ExactWBLearn with probability at least $1 - 3n^{-7}$ over the distribution of $\mathbf{P}$. We next use ideas from [26, 6] to show this implies that Learner solves $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$ in the worst case.

Indeed, assume that Learner fails on some instance P of $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$. By the promise of the problem, there exists a circuit $\widehat{C}$ such that $cd^t(P, \widehat{C}) \leq 2 \log n$, and $\widehat{C}$ agrees with P with probability at least $1 - n^{-10}$. Let $\ell = \mathrm{K}^t(P, \widehat{C})$. Our goal is to show that $\mathrm{K}(P, \widehat{C}) < \ell - 2 \log n$, which is a contradiction.

Toward this goal, let Let $\mathcal{S}_\ell$ be the set of all pairs of circuits $(P', \widehat{C'})$ with $\mathrm{K}^t(P', \widehat{C'}) = \ell$ that agree with probability at least $1 - n^{-10}$, and on which Learner fails, so that $(P, \widehat{C}) \in \mathcal{S}_\ell$. Fix $(P', \widehat{C'}) \in \mathcal{S}_\ell$, and let Π be the length ℓ program that outputs $(P', \widehat{C'})$ in time t. Observe that the probability of A to sample $(P', \widehat{C'})$ in the first step of the above protocol is at least its probability to sample the program Π, which is $1/2n \cdot 2^{-\ell}$. Since $(P', \widehat{C'})$ agree with high probability, the equality test it the third step of the protocol will pass with high probability, and thus A will send P' to B with probability at least $1/4n \cdot 2^{-\ell}$. In other words,

$$\Pr[\mathbf{P} = P'] \geq 1/4n \cdot 2^{-\ell}$$

for every $(P', \widehat{C'}) \in \mathcal{S}_\ell$, and thus

$$\Pr[(\mathbf{P}, \cdot) \in \mathcal{S}_\ell] \geq |\mathcal{S}_\ell| \cdot 1/4n \cdot 2^{-\ell}$$

(here we say that $(P, \cdot) \in \mathcal{S}_\ell$ if there is some C such that $(P, C) \in \mathcal{S}_\ell$).

On the other hand, by definition of $\mathcal{S}_\ell$, Learner fails on every $(P', \widehat{C'}) \in \mathcal{S}_\ell$. Since Learner fails with probability at most $3n^{-7}$ on $\mathbf{P}$, it must holds that

$$3n^{-7} \geq \Pr[(\mathbf{P}, \cdot) \in \mathcal{S}_\ell].$$

[9] This particular non-black usage of Eve is not inherent. We could have considered a different formalization of the learning theory problem which simply requires the attacker to succeed on a randomly sampled instance. Subsequent parts of the argument, however, will use non-black-box access to Eve more inherently.

Combining the above, we get that

$$|\mathcal{S}_\ell| \leq 12n^{-6} \cdot 2^\ell.$$

We can now use the bound on $\mathcal{S}_\ell$ to bound the Kolmogorov complexity of $(P, \widehat{C})$. To describe $(P, \widehat{C})$, it is enough to describe the set $\mathcal{S}_\ell$, and the index of the pair $(P, \widehat{C})$ in this set. That is,

$$\mathrm{K}(P, \widehat{C}) \leq \mathrm{K}(\mathcal{S}_\ell) + \log|\mathcal{S}_\ell| + O(\log\log n) \leq \mathrm{K}(\mathcal{S}_\ell) + \ell - 6\log n + O(\log\log n).$$

We conclude the proof with the observation that to describe $\mathcal{S}_\ell$ it is enough to describe n (which can be done using $\log n + O(\log\log n)$ bits), ℓ ($\log n$ bits) and Eve (that can be described with constant many bits[10]). Thus, $\mathrm{K}(\mathcal{S}_\ell) \leq 3\log n$, and we get that

$$\mathrm{K}(P, \widehat{C}) < \ell - 2\log n,$$

as we wanted to show.

Hardness of ExactWBLearn$|_{\mathsf{CS}^t}$ from PKE

We next show that the existence of PKE implies the hardness of $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$. We now sketch the proof. Let Gen be the algorithm the generate pair (pk, sk) of public and secret key, and let $\mathsf{Enc}, \mathsf{Dec}$ be the encryption and decryption protocols. For a random pair of keys $(pk, sk) \leftarrow \mathsf{Gen}(r)$, we construct two circuits, $P(w, s) = (x = \mathsf{Enc}_{pk}(s; w), s)$, and $\widehat{C}(x) = \mathsf{Dec}_{sk}(x)$.

By the security of the PKE scheme, it follows that with high probability over the randomness of Gen, it is hard to learn the function of C, as the circuit P only uses the public key, and the function C computes is the decryption of a random encryption. This already implies that $\mathsf{ExactWBLearn}$ is hard, but we still need to show that P is inside the promise of the problem $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$.

It follows by the correctness of the PKE scheme that $\widehat{C}$ computes the function that is sampled by P. Thus, to be in the promise of $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$, we only need that $(P, \widehat{C})$ has small computational depth. This, however, is not necessarily the case. To solve this, we simply pad $\widehat{C}$ with the randomness used by Gen. That is, we let $\widehat{C'}$ be a functionally equivalent circuit to $\widehat{C}$, with r encoded to it, where r is such that $\mathsf{Gen}(r) = (pk, sk)$. It follows that when t is large enough, $\mathrm{K}^t(P, \widehat{C'}) \leq |r| + O(1)$ (as we can describe them by simply describing the randomness and the algorithm Gen). On the other hand, $\mathrm{K}(P, \widehat{C'}) \geq \mathrm{K}(r)$ (since r can be obtained from $\widehat{C'}$), which is at least $|r| - O(1)$ with high probability. Together, we conclude that with high probability over the randomness of Gen, the circuits $(P, \widehat{C'})$ are in the promise of $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$.

Comparison with [6]

We remark that at a high-level, our construction of a two-message key-agreement protocol shares many similaries with the key-agreement protocol developing in [6], relying on the hardness of an interactive notion of time-bounded Kolmogorov complexity, conditionned on an analog of computational shallowness. They central difference is that the protocol in [6] requires at least 3 rounds due to the use of an equility check protocol to determine whether

[10] This non-black-box usage of Eve (which is taken from [26, 6]) is seemingly inherent to our proof technique. Note that we are here relying on the fact that Eve is a uniform algorithm, but as we discuss in the formal section, the argument can be extended to work also in the non-uniform setting.

Alice and Bob managed to agree on a key. In contrast, our protocol does not rely on such an equality check step (and thus it can be executed in only two rounds, which is crucial to get PKE); indeed, we replaced the equality protocol with a step where Alice on her own can determine whether the message she sends will enable agreement to happen. Of course, the reason why we can do this is that we are reducing security from a different problem.

Other features of the protocol are quite similar; this is because we also rely on the worst-case hardness of a problem "conditioning on computationally shallow instances". Indeed, as described above, our security proof shares many features with those of [25, 6].

Comparison with Universal Constructions

Universal constructions of PKE are known (see [16]); that is, constructions having the property that they are secure if (any) PKE exist. We emphasize that while the details of the protocol from [16] are somewhat different, the "agreement estimation" step performed there is very similar to what we do. Furthermore, we can interpret our protocol as an alternative (variant) universal PKE protocol in which Alice chooses a random (key-generation) program Gen and executes it to get an encryption scheme Enc_{pk} and description scheme Dec_{sk}, with the keys pk, sk hardcoded to the scheme.[11] Alice then estimates the agreement probability of the encryption, and if it is high enough, she sends the encryption scheme as the public-key, and uses the decryption scheme as the private-key. If PKE exists, with a noticeable probability over the choice of Gen, this scheme will be secure. We emphasize that since we base the security of our protocol (and thus also the above universal one) on the hardness of the white-box learning problem, it enables an approach for measuring the concrete security of the protocol by relating it to the security of the learning theory problem.

2 Preliminaries

2.1 Notations

All logarithms are taken in base 2. We use calligraphic letters to denote sets and distributions, bold uppercase for random variables, and lowercase for values and functions. Let poly stand for the set of all polynomials. Let PPT stand for probabilistic poly-time, and n.u.-poly-time stand for non-uniform poly-time. An n.u.-poly-time algorithm A is equipped with a (fixed) poly-size advice string set $\{z_n\}_{n\in\mathbb{N}}$ (that we typically omit from the notation), and we let A_n stand for A equipped with the advice z_n (used for inputs of length n). For a randomized algorithm A, we denote by $\mathsf{A}(\cdot; r)$ the algorithm A with fixed randomness $r \in \{0,1\}^*$. Let neg stand for a negligible function. Given a vector $v \in \Sigma^n$, let v_i denote its i^{th} entry, let $v_{<i} = (v_1, \ldots, v_{i-1})$ and $v_{\leq i} = (v_1, \ldots, v_i)$. Similarly, for a set $\mathcal{I} \subseteq [n]$, let $v_{\mathcal{I}}$ be the ordered sequence $(v_i)_{i\in\mathcal{I}}$. For $x, y \in \{0,1\}^*$, we use $x||y$ to denote the concatenation of x and y. For a set $\mathcal{S} \subseteq \{0,1\}^*$, we use $\mathcal{S}||y$ to denote the set $\{x||y\colon x \in \mathcal{S}\}$.

2.2 Distributions and Random Variables

When unambiguous, we will naturally view a random variable as its marginal distribution. The support of a finite distribution $\mathcal{P}$ is defined by $\text{Supp}(\mathcal{P}) := \{x\colon \Pr_{\mathcal{P}}[x] > 0\}$. For a (discrete) distribution $\mathcal{P}$, let $x \leftarrow \mathcal{P}$ denote that x was sampled according to $\mathcal{P}$. Similarly, for a set $\mathcal{S}$, let $x \leftarrow \mathcal{S}$ denote that x is drawn uniformly from $\mathcal{S}$. For $m \in \mathbb{N}$, we use $\mathbf{U}_m$ to denote

[11] In the protocol of [16], Alice would instead choose random programs for Alice and Bob to run; we do not know how to prove the security of such a protocol under our assumption.

a uniform random variable over $\{0,1\}^m$ (that is independent from other random variables in consideration). The statistical distance (also known as, variation distance) of two distributions $\mathcal{P}$ and $\mathcal{Q}$ over a discrete domain $\mathcal{X}$ is defined by $\mathrm{SD}(\mathcal{P},\mathcal{Q}) := \max_{\mathcal{S}\subseteq\mathcal{X}}|\mathcal{P}(\mathcal{S}) - \mathcal{Q}(\mathcal{S})| = \frac{1}{2}\sum_{x\in\mathcal{S}}|\mathcal{P}(x) - \mathcal{Q}(x)|$.

The following lemma is proven in the full version of this paper.

▶ **Lemma 8.** *There exists an efficient oracle-aided algorithm* A *such that the following holds. Let* $(\mathbf{X},\mathbf{Y})$ *be a pair of jointly distributed random variables over* $\{0,1\}^* \times \{0,1\}^n$*, and let* $\mathbf{R}$ *be a uniform independent random variable over* $\{0,1\}^n$*. Let* E *be an algorithm such that* $\Pr[\mathsf{E}(\mathbf{X},\mathbf{R}) = \langle \mathbf{Y},\mathbf{R}\rangle] \geq 1-\epsilon$*, for* $0 \leq \epsilon$*. Then* $\Pr\left[\mathsf{A}^{\mathsf{E}}(1^n,\mathbf{X}) = \mathbf{Y}\right] \geq 1 - 8\epsilon - \mathrm{neg}(n)$.

2.3 Circuits

In this paper we consider circuits over the De-Morgan Basis, which contains the following gates: $\wedge$ ("and" gate with fan-in 2), $\vee$ ("or" gate with fan-in 2), and $\neg$ ("not" gate with fan-in one). The size of a circuit C, is the number of gates in C.

We consider encoding of a circuit as a string over $\{0,1\}^*$ in the following natural way: Given a circuit C, we first encode the length of C using a prefix-free encoding, and then for every gate g, according to a topological order, we encode its type (input, output, $\wedge$,$\vee$, or $\neg$), and the (up to two) gates wired into g.

Observe that every circuit C of size s can be encoded to a string of length $O(s\log s)$. Moreover, given the encoding and an input x to C, $C(x)$ can be computed efficiently (in polynomial time in the encoding length).

We identify a string in $\{0,1\}^*$ with the circuit it encodes. For example, for $C \in \{0,1\}^n$ we use $C(x)$ to denote the value of the circuit encoded by C applied on the input x. We denote by $|C|$ the length of the encoding of C.

Finally, by encoding first the size of C, we assume that for every encoding of a circuit C, and for every $z \in \{0,1\}^*$, the string $C||z$ encodes the same circuit as C.[12]

2.4 Entropy

For a random variable $\mathbf{X}$, let $\mathrm{H}(\mathbf{X}) = \mathrm{E}[\log \frac{1}{\Pr[\mathbf{X}=x]}]$ denote the (Shannon) entropy of $\mathbf{X}$, and let $\mathrm{H}_\infty(\mathbf{X}) = \min_{x\in Supp(\mathbf{X})} \log \frac{1}{\Pr[\mathbf{X}=x]}$ denote the *min-entropy* of $\mathbf{X}$.

$$\min_{x\in \mathrm{Supp}(\mathbf{X})} \log \frac{1}{\Pr[\mathbf{X}=x]}.$$

For a random variable $\mathbf{X}$ and an event E, we use $\mathrm{H}_\infty(\mathbf{X} \mid E)$ to denote the min-entropy of the distribution $\mathbf{X}|_E$. We will use the following facts.

▶ **Fact 9.** *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be independent random variables. Then* $\mathrm{H}_\infty(\mathbf{X},\mathbf{Y}) = \mathrm{H}_\infty(\mathbf{X}) + \mathrm{H}_\infty(\mathbf{Y})$.

2.5 Complexity Classes

We define the complexity classes FBPP and ioP/poly.

[12] We actually only use the fact that we can encode z into the circuit. This can be done by adding dummy gates that do not change the output of C, where each gate g_i is either $\wedge$ or $\vee$ according to the ith bit of z.

▶ **Definition 10** (Infinitely-often FBPP (ioFBPP)). *A binary relation $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ is in* ioFBPP *if there exists* PPT *algorithms* A *such that the following holds for infinitely many n's:*

For every $x \in \{0,1\}^n$ such that there exists $y \in \{0,1\}^*$ with $(x,y) \in \mathcal{R}$,

$$\Pr\left[(x, \mathsf{A}(x, 1^k)) \in \mathcal{R}\right] \geq 1 - 1/k,$$

for every $k > 0$. $\mathcal{R}$ is in ioP/poly *if the above holds with respect to $n.u. - poly - time$ algorithms* A*.*

2.6 One-Way Function

▶ **Definition 11** (One-way function). *A polynomial-time computable function $f : \{0,1\}^* \to \{0,1\}^*$ is called a* one-way function *if for every polynomial-time algorithm* A*,*

$$\Pr_{x \leftarrow \{0,1\}^n}\left[\mathsf{A}(1^n, f(x)) \in f^{-1}(f(x))\right] = \mathrm{neg}(n)$$

▶ **Definition 12** (Weak one-way function). *Let $m \in$ poly be a polynomial-time computable function. A polynomial-time computable function $f : \{0,1\}^{m(n)} \to \{0,1\}^*$ is called α-*weak one-way function *if for every polynomial-time algorithm* A*, for every large enough n,*

$$\Pr_{x \leftarrow \{0,1\}^{m(n)}}\left[\mathsf{A}(1^n, f(x)) \in f^{-1}(f(x))\right] \leq 1 - \alpha(n)$$

f is a weak one-way function *if it is $1/p$-weak one way function, for some $p \in$ poly.*

▶ **Theorem 13** (Weak to strong OWFs, [36]). *One-way functions exist if and only if weak-one way functions exist.*

2.7 Public-Key Encryption

▶ **Definition 14** (Public-key encryption scheme (PKE)). *A triplet of randomized, efficiently computable functions* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a* $(\alpha(n), \beta(n))$-public-key encryption scheme (PKE) *if the following holds:*

- *Correctness: For every large enough $n \in \mathbb{N}$ and any $m \in \{0,1\}$,*

$$\Pr_{(sk,pk) \leftarrow \mathsf{Gen}(1^n)}[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m] \geq 1 - \alpha(n)$$

- *Security: For every* PPT Eve*, for every large enough $n \in \mathbb{N}$,*

$$\Pr_{(sk,pk) \leftarrow \mathsf{Gen}(1^n), m \leftarrow \{0,1\}}[\mathsf{Eve}(pk, \mathsf{Enc}(pk, m)) = m] \leq 1/2 + \beta(n).$$

Such a scheme is a PKE *if it is $(1/n^c, 1/n^c)$-PKE for every $c \in \mathbb{N}$.*

The following lemma shows that it is possible to amplify an 1-bit weak key-agreement protocol into a key-agreement. This lemma is a simple case of the more general result of [19].

▶ **Lemma 15** (PKE amplification, [19]). *The following holds for every constants $c_1 > c_2$. Assume there exists an $(n^{-c_1}, 1/2 - n^{-c_2})$-PKE. Then, there exists a PKE.*

We also define weak-PKE.

▶ **Definition 16** (Weak Public-key encryption scheme (weak-PKE)). *For an efficiently computable function d: $\{1\}^* \to \mathbb{N}$, a triplet of randomized, efficiently computable functions* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a* $(\alpha(n), \beta(n), \gamma(n))$-weak-public-key encryption scheme (weak-PKE) *if the following holds:*

- *For every $n \in \mathbb{N}$, given pk and randomness r,* $\mathsf{Enc}(pk; r)$ *outputs a message $m(pk; r)$ and an output $o(pk; r) \in \mathbb{N}$.*
- *Correctness: For every large enough $n \in \mathbb{N}$*

$$\Pr_{(sk,pk)\leftarrow \mathsf{Gen}(1^n),r}[|\mathsf{Dec}(sk, m(pk, r)) - o(pk, r)| \leq d(1^n)] \geq 1 - \alpha(n)$$

- *Security: For every* PPT Eve*, for every large enough $n \in \mathbb{N}$,*

$$\Pr_{(sk,pk)\leftarrow \mathsf{Gen}(1^n),r}[|\mathsf{Eve}(pk, m(pk, r)) - o(pk, r)| \leq \gamma(n) \cdot d(1^n)] \leq \beta(n).$$

In the full version of this paper we prove the following lemma, stating that weak-PKE can be used to construct PKE.

▶ **Lemma 17** (Weak-PKE amplification)**.** *The following holds for every constants $c_1 > c_2$. Assume there exists an $(n^{-c_1}/2, 1 - 10n^{-c_2}, 2n^{c_1})$-weak-PKE. Then, there exists a PKE.*

2.8 Kolmogorov Complexity and Computational Depth

Roughly speaking, the *t-time-bounded Kolmogorov complexity*, $\mathrm{K}^t(x)$, of a string $x \in \{0,1\}^*$ is the length of the shortest program $\Pi = (M, y)$ such that, when simulated by an universal Turing machine, Π outputs x in $t(|x|)$ steps. Here, a program Π is simply a pair of a Turing Machine M and an input y, where the output of P is defined as the output of $M(y)$. When there is no running time bound (i.e., the program can run in an arbitrary number of steps), we obtain the notion of Kolmogorov complexity.

In the following, fix universal TM U with polynomial simulation overhead, and let $\mathsf{U}(\Pi, 1^t)$ denote the output of Π when emulated on U for t steps. We now define the notion of Kolmogorov complexity with respect to the universal TM U.

▶ **Definition 18.** *Let t be a polynomial. For all $x \in \{0,1\}^*$, define the* t-bounded Kolmogorov complexity *of x*

$$\mathrm{K}^t(x) = \min_{\Pi \in \{0,1\}^*} \{|\Pi| : \mathsf{U}(\Pi, 1^{t(|x|)}) = x\}$$

where $|\Pi|$ is referred to as the description length *of Π. When there is no time bound, the* Kolmogorov complexity *of x is defined as*

$$\mathrm{K}(x) = \min_{\Pi \in \{0,1\}^*} \{|\Pi| : \exists t \in \mathbb{N} \mathsf{U}(\Pi, 1^t) = x\}.$$

The computational depth *of x [4], denoted by $cd^{t,\infty}(x)$, is defined to be*

$$cd^{t,\infty}(x) = \mathrm{K}^t(x) - \mathrm{K}(x).$$

We use $\mathsf{K}(x, y)$ to denote the Kolmogorov complexity of some generic self-delimiting encoding of the pair x, y. Recall that we use $\mathsf{K}(x||y)$ to denote the complexity of the concatenation of x and y. We will use the following well-known fact:

▶ **Fact 19.** *For every $x, y \in \{0,1\}^*$,*

$$\mathsf{K}(x, y) \leq \mathsf{K}(x) + \mathsf{K}(y) + \log(\mathsf{K}(x)) + 2\log\log(\mathsf{K}(x)) + O(1).$$

We will also use the following bound on the Kolmogorov complexity of strings sampled from distributions with high min-entropy.

▶ **Lemma 20.** *For every $n \in \mathbb{N}$, and every distribution $\mathcal{D}$, it holds that*

$$\Pr_{x \leftarrow \mathcal{D}}[\mathsf{K}(x) \geq \mathrm{H}_\infty(\mathcal{D}) - \log n] \geq 1 - 1/n.$$

Proof. There are at most $2^{\mathrm{H}_\infty(\mathcal{D}) - \log n}$ strings x with $\mathsf{K}(x) < \mathrm{H}_\infty(\mathcal{D}) - \log n$. Thus,

$$\Pr_{x \leftarrow \mathcal{D}}[\mathsf{K}(x) < \mathrm{H}_\infty(\mathcal{D}) - \log n] \leq 2^{\mathrm{H}_\infty(\mathcal{D}) - \log n} \cdot 2^{-\mathrm{H}_\infty(\mathcal{D})} = 1/n. \qquad \blacktriangleleft$$

We will also use the well-known Chernoff bound in our proof.

▶ **Fact 21** (Hoeffding's inequality). *Let $\mathbf{A}_1, ..., \mathbf{A}_n$ be independent random variables s.t. $\mathbf{A}_i \in \{0,1\}$. Let $\widehat{\mathbf{A}} = 1/n \cdot \Sigma_{i=1}^n \mathbf{A}_i$ and $\mu = \mathrm{E}\left[\widehat{\mathbf{A}}\right]$. For every $\epsilon \in [0,1]$ it holds that:*

$$\Pr\left[\left|\widehat{\mathbf{A}} - \mu\right| \geq \epsilon\right] \leq 2 \cdot e^{-\epsilon^2 \cdot n}.$$

3 White-Box Distributional Learning

Let $P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell$ be a circuit that, given r bits of randomness, samples labeled instances (x,s). In the following we view s as a binary representation of a number in $\mathbb{N}$ (and respectively all the operations below are over $\mathbb{N}$, and we use $|\cdot|$ to denote the absolute value). We define the set of all circuits that approximate P with high probability,

$$\mathsf{Comp}_\epsilon^\Delta(P) = \left\{C\colon \{0,1\}^k \to \{0,1\}^\ell \colon \Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|C(x) - s| \leq \Delta] \geq 1 - \epsilon\right\}.$$

We define the following white-box learning problem $\mathsf{WBLearn}$:

▶ **Definition 22** (Δ-$\mathsf{WBLearn}$). *For a function $\Delta\colon \{1\}^* \to \mathbb{N}$, let Δ-$\mathsf{WBLearn}$ be the following learning problem:*

- *Input: Circuit $P \in \{0,1\}^n$, with the promise that there exists a circuit $\widehat{C} \in \{0,1\}^n$ such that $\widehat{C} \in \mathsf{Comp}_{n^{-10}}^{\Delta(1^n)/n^{10}}(P)$.*
- *Output: Circuit $C \in \mathsf{Comp}_{1/3}^{\Delta(1^n)}(P)$*

In this work we are focusing on inputs P for which the circuit $\widehat{C}$ that agree with P with high probability, is such that the description of $\widehat{C}$ and P is computational shallow. Formally, for a time function $t\colon \mathbb{N} \to \mathbb{N}$, we denote by Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t}$ the problem Δ-$\mathsf{WBLearn}$ with the additional promise that $cd^t(P, \widehat{C}) \leq 2 \log n$.

We use $\mathsf{ExactWBLearn}$ and $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$ to denote Δ-$\mathsf{WBLearn}$ and Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t}$ when $\Delta(1^n) = 0$. Note that $\mathsf{ExactWBLearn}$ and Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t}$ are incomparable: While in Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t}$ we only need to find a circuit that approximates P, the promise in $\mathsf{ExactWBLearn}$ is stronger. Yet, there is a simple reduction from $\mathsf{ExactWBLearn}$ to Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t}$.

▶ **Lemma 23.** *For every $\Delta\colon \{1\}^* \to \mathbb{N}$ such that $\Delta \in 2^{o(n/\log n)}$, it holds that*

$$\mathsf{ExactWBLearn} \leq_p \Delta\text{-}\mathsf{WBLearn}.$$

Similarly, for any such Δ and a function $t\colon \mathbb{N} \to \mathbb{N}$, there exists $t'\colon \mathbb{N} \to \mathbb{N}$, such that

$$\mathsf{ExactWBLearn}|_{\mathsf{CS}^t} \leq_p \Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^{t'}}.$$

Proof. We start with the first reduction $\mathsf{ExactWBLearn} \leq_p \Delta\text{-}\mathsf{WBLearn}$. Given a circuit $P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell$ of length n, the reduction outputs a circuit $P'\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^{\ell+n}$, which is equivalent to P, with additional n output gates that always output 0. That is, $P'(w) = (x, s')$, where $P(w) = (x, s)$ and $s' = 2^n \cdot s$. As we only added n gates to P, P' can be encoded using $O(n \log n)$ bits. Using padding we assume that $|P'| \in \Theta(n \log n)$.

For correctness, observe that if there exists $\widehat{C} \in \mathsf{Comp}^0_{n^{-10}}(P)$, then there exists such $\widehat{C'} \in \mathsf{Comp}^0_{n^{-10}}(P')$ (where $\widehat{C'}$ is defined similarly to P' using $\widehat{C}$). Moreover, an approximation of the output s' of P' within a distance of $2^n/4$ is equivalent to the exact output s of P. Indeed, given an $2^n/4$-approximation of s' we can find s by simply dividing s' by 2^n and rounding to the closest integer.

Finally, to see the second reduction, it is enough to show that $cd^{t'}(P', \widehat{C'}) \leq 2\log|P'|$. Since $P', \widehat{C'}$ can be efficiently constructed given $P, \widehat{C}$, and similarly $P, \widehat{C}$ can be efficiently constructed given $P', \widehat{C'}$, it holds that for large enough (polynomial time t', $cd^{t'}(P', \widehat{C'}) \leq cd^t(P, \widehat{C}) + O(1) \leq 2\log n + O(1) \leq 2\log n + 2\log\log n \leq 2\log|P'|$. ◀

We prove the following theorem.

▶ **Theorem 24.** *Let $\epsilon > 0$ be a constant and let $\Delta\colon \{1\}^* \to \mathbb{N}$ be any efficiently computable function such that $\Delta(1^n) \leq 2^{n^{(1-\epsilon)}}$, and let $t\colon \mathbb{N} \to \mathbb{N}$ be any polynomial such that $t(n) \geq n^{1+\epsilon}$. Then the following are equivalent:*

- *PKE exists*
- $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$

As a corollary, we get a result of independent interest relating exact and approximate white-box learning:

▶ **Corollary 25.** *Let $\epsilon > 0$ be a constant and let $\Delta\colon \{1\}^* \to \mathbb{N}$ be any efficiently computable function such that $\Delta(1^n) \leq 2^{n^{(1-\epsilon)}}$, and let $t\colon \mathbb{N} \to \mathbb{N}$ be any polynomial such that $t(n) \geq n^{1+\epsilon}$. Then the following are equivalent:*

- $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$
- $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$

Theorem 24 follows by the following two theorems.

▶ **Theorem 26.** *Let $t(n)$ be a polynomial and $\Delta\colon \{1\}^* \to \mathbb{N}$ an efficiently computable function. Then if $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$, PKE exist.*

▶ **Theorem 27.** *Assume there exists a PKE. Then for any constant $\epsilon > 0$ and any $t(n) \geq n^{1+\epsilon}$ and any efficiently computable $\Delta\colon \{1\}^* \to \mathbb{N}$ such that $\Delta(1^n) \leq 2^{n^{(1-\epsilon)}}$, $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$.*

Theorem 26 is proven in Section 4, and Theorem 27 is proven in the full version of this paper.

4 Worst-Case hardness of $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t} \implies \mathsf{KE}$

In this section we prove Theorem 26, that states that the worst-case hardness of $\mathsf{WBLearn}|_{\mathcal{Q}_t}$ implies the existence of public-key encryption scheme. In the following, let $C_0\colon \{0,1\} \to \{0,1\}$ be the circuit that always outputs 0, and $P_0\colon \{0,1\} \to \{0,1\} \times \{0,1\}$ be the circuit that always outputs $(0,0)$.

To prove the above theorem, fix $t = t(n)$ and $\Delta = \Delta(1^n)$, and consider the following scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$:

▶ **Algorithm 28** (Gen).
Parameter: function $t\colon \mathbb{N} \to \mathbb{N}$.
Input: 1^n.
Operation:
1. *Sample* $\lambda \leftarrow [3n]$ *and* $\Pi \leftarrow \{0,1\}^\lambda$.
2. *Run* Π *for* $t(2n)$ *steps to get circuits* $C\colon \{0,1\}^k \to \{0,1\}^\ell, P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell$ *for some* k, r, ℓ. *If the output of* Π *is not two such circuits, set* $C = C_0$, $P = P_0$, $r = k = \ell = 1$.
3. *Randomly sample* $(x_1, s_1), \ldots, (x_{n^{20}}, s_{n^{20}}) \leftarrow P(U_r)$, *and compute*

$$\widehat{\alpha} = \Pr_{i \leftarrow [n^{20}]}\left[|C(x_i) - s_i| \leq \Delta(1^n)/n^{10}\right].$$

   If $\widehat{\alpha} < 1 - 2n^{-8}$, *reset* $C = C_0$ *and* $P = P_0$.
4. *Output* (k, ℓ, C) *as the secret key, and* (r, k, ℓ, P) *as the public key.*

▶ **Algorithm 29** (Enc).
Input: public-key $(r \in \mathbb{N}, k \in \mathbb{N}, \ell \in \mathbb{N}, P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell)$.
Operation:
1. *Sample randomness* $z \leftarrow \{0,1\}^r$.
2. *Compute* $P(z)$ *to get* (x, s).
3. *Output* $m = x$ *and* $o = s$.

▶ **Algorithm 30** (Dec).
Input: secret-key $(k \in \mathbb{N}, \ell \in \mathbb{N}, C\colon \{0,1\}^k \to \{0,1\})$, *cipher* $x \in \{0,1\}^k$.
Operation:
1. *Compute* $C(x) = s'$.
2. *Output* s'.

Observe that the size of the circuits C and P sampled by $\mathsf{Gen}(1^n)$ is at most $t(2n)$. Thus, all of the above algorithms can be implemented efficiently. Below we bound the correctness and the security of the above scheme. For every $n \in \mathbb{N}$, let $(\mathbf{K}_n, \mathbf{L}_n, \mathbf{C}_n)$ and $(\mathbf{R}_n, \mathbf{K}_n, \mathbf{L}_n, \mathbf{P}_n)$ be the random variables distributed according to the secret and public keys $(k, \ell, C), (r, k, \ell, P)$ in a random execution of $\mathsf{Gen}(1^n)$. Let $\mathbf{M}_n$ and $\mathbf{O}_n$ be the output of $\mathsf{Enc}(\mathbf{R}_n, \mathbf{K}_n, \mathbf{L}_n, \mathbf{P}_n)$ in a random execution.

4.1 Correctness

We start by analyzing the correctness probability of the scheme.

▶ **Lemma 31.** *For every* $n \in \mathbb{N}$, *it holds that*

$$\Pr\left[|\mathsf{Dec}((\mathbf{K}_n, \mathbf{L}_n, \mathbf{C}_n), \mathbf{M_n}) - \mathbf{O}_n| \leq \Delta(1^n)/n^{10}\right] \geq 1 - n^{-7}.$$

To prove Lemma 31 we will use the following simple claim, which is immediate from the Hoffeding bound.

▷ Claim 32. Let $C\colon \{0,1\}^k \to \{0,1\}^\ell$ and $P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell$ be two circuits. Let $\mathbf{A}$ be a random variable distributed according to the following process:

Sample $(x_1, s_1) \ldots, s(x_{n^{20}}, s_{n^{20}}) \leftarrow P(\mathbf{U}_r)$, and let $\mathbf{A} = \Pr_{i \leftarrow [n^{20}]}[|C(x_i) - s_i| \leq \Delta]$.
Let $\mu = \Pr_{(x,s) \leftarrow P(\mathbf{U}_r)}[|C(x) - s| \leq \Delta]$. Then $\Pr\left[|\mathbf{A} - \mu| \geq n^{-9}\right] \leq 2^{-n}$.

Proof. Immediate by Fact 21. ◁

Proof of Lemma 31. When $\mathsf{Gen}(1^n)$ outputs C_0 and P_0, the scheme has perfect correctness. Moreover, when the secret and public key are (k, ℓ, C) and (r, k, ℓ, P), it holds that

$$\Pr\left[|\mathsf{Dec}((k,\ell,C),\mathbf{M})) - \mathbf{O}| \leq \Delta(1^n)/n^{10}\right] = \Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}\left[|C(x) - s| \leq \Delta(1^n)/n^{10}\right].$$

Thus,

$$\begin{aligned}
&\Pr\left[|\mathsf{Dec}((\mathbf{K}_n,\mathbf{L}_n,\mathbf{C}_n),\mathbf{M}_n) - \mathbf{O}_n| \leq \Delta(1^n)/n^{10}\right] \\
&\geq (1-3n^{-8}) \cdot \Pr_{\mathbf{R}_n,\mathbf{K}_n,\mathbf{P}_n,\mathbf{C}_n}\left[\Pr_{(x,s)\leftarrow \mathbf{P}_n(\mathbf{U}_{\mathbf{R}_n})}\left[|\mathbf{C}_n(x) - s| \leq \Delta(1^n)/n^{10}\right] \geq 1-3n^{-8}\right] \\
&\geq 1 - 3n^{-8} - \Pr_{\mathbf{R}_n,\mathbf{K}_n,\mathbf{P}_n,\mathbf{C}_n}\left[\Pr_{(x,s)\leftarrow \mathbf{P}_n(\mathbf{U}_{\mathbf{R}_n})}\left[|\mathbf{C}_n(x) - s| \leq \Delta(1^n)/n^{10}\right] < 1-3n^{-8}\right].
\end{aligned}$$

The lemma now follows since by Lemma 31, the probability of circuits P, C with

$$\Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}\left[|\mathbf{C}_n(x) - s| \leq \Delta(1^n)/n^{10}\right] < 1 - 3n^{-8}$$

to pass the test in Step 3 is at most 2^{-n}. ◀

4.2 Security

We next bound the leakage of the scheme.

▶ **Lemma 33.** *Assume there exists an algorithm* Eve *such that*

$$\Pr[|\mathsf{Eve}(1^n,(\mathbf{R}_n,\mathbf{K}_n,\mathbf{L}_n,\mathbf{P}_n),\mathbf{M_n}) - \mathbf{O}_n| \leq \Delta(1^n)] \geq 1 - n^{-6}$$

for infinitely many n's. Then Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t} \in \mathsf{ioBPP}$.

In the following, let Eve be an algorithm that uses $r_{\mathsf{Eve}}(n)$ bits of randomness and guesses $\mathbf{M}$ with probability at least $1-n^{-6}$. Recall that for $w \in \{0,1\}^{r_{\mathsf{Eve}}(n)}$, $\mathsf{Eve}(1^n,(r,k,\ell,P),m;w)$ denotes the execution of Eve when its randomness is fixed to be w.

▶ **Algorithm 34.**
Input: $P \in \{0,1\}^n$.
Operation:
1. *Let r, k, ℓ be such that $P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell$.*
2. *Sample $w \leftarrow \{0,1\}^{r_{\mathsf{Eve}}(n)}$ uniformly at random.*
3. *Construct a circuit C' such that $C'(x) = \mathsf{Eve}(1^n,(r,k,\ell,P),x;w)$.*
4. *Return C'.*

We prove the following lemma.

▶ **Lemma 35.** *Assume that*

$$\Pr[|\mathsf{Eve}(1^n,(\mathbf{R}_n,\mathbf{K}_n,\mathbf{L}_n,\mathbf{P}_n),\mathbf{M}_n) - \mathbf{O}_n| \leq \Delta(1^n)] \geq 1 - n^{-6}$$

for infinitely many n's. Then the following holds for infinitely many n's. For every input $P \in \{0,1\}^n \cap \mathcal{Q}_t^\Delta$, Algorithm 34 outputs $C \in \mathsf{Comp}_{1/4}^\Delta(P)$ with probability at least $1/2$.

We prove Lemma 35 below, but first we use it to prove Lemma 33.

Proof of Lemma 33. Assume there exists an algorithm Eve such that

$$\Pr[|\mathsf{Eve}(1^n, (\mathbf{R}_n, \mathbf{K}_n, \mathbf{L}_n, \mathbf{P}_n), \mathbf{M}_n) - \mathbf{O}_n| \leq \Delta(1^n)] \geq 1 - n^{-6}$$

for infinitely many n's.

By Lemma 35, for infinitely many n's, Algorithm 34 outputs $C \in \mathsf{Comp}^{\Delta}_{1/4}(P)$ with probability at least $1/2$, for every $P \in \{0,1\}^n \cap \mathcal{Q}^{\Delta}_t$. We want to construct an algorithm Sol that, given P and $1^{1/\delta}$, outputs $C \in \mathsf{Comp}_{1/3}(P)$ with probability at least $1-\delta$, for every such n.

Let Sol be the algorithm that, given input $(P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell, 1^{1/\delta})$, first run Algorithm 34 on P for $2\lceil \log 1/\delta \rceil$ times, to get circuits $C_1, \ldots, C_{2\lceil \log 1/\delta \rceil}$. Then, for every circuit C_i, Sol samples $(x_1, s_1), \ldots, (x_{(100\lceil \log 1/\delta \rceil)^2}, s_{(100\lceil \log 1/\delta \rceil)^2}) \leftarrow P(\mathbf{U}_r)$, and computes

$$p_i = \Pr_{j \leftarrow [(100\lceil \log 1/\delta \rceil)^2]}[|C_i(x_j) - s_j| \leq \Delta(1^n)].$$

Finally, Sol outputs the circuit C_i for the index i with maximal value of p_i.

We now analyze the success probability of Sol. Using *Fact* 21, for every i, the probability that $\left|p_i - \Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|C_i(x) = s| \leq \Delta(1^n)]\right| \geq 1/25$ is at most

$$2^{-4(\lceil \log 1/\delta \rceil)^2 + 1} \leq 1/(\delta(4\lceil \log 1/\delta \rceil)).$$

Thus, by the union bound, the probability that $\left|p_i - \Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|C_i(x) = s| \leq \Delta(1^n)]\right| \geq 1/25$ for some i is at most $\delta/2$. Next, by the success probability of Algorithm 34, with probability at least

$$1 - (1 - 1/2)^{2\log 1/\delta} \geq 1 - \delta/2,$$

at least one of the circuits $C_1, \ldots, C_{2\log 1/\delta}$ is in $\mathsf{Comp}^{\Delta}_{1/4}(P) \subseteq \mathsf{Comp}^{\Delta}_{1/3}(P)$. Let i^* be the index of such a circuit. Then, with probability at least $1 - \delta/2 - \delta/2 = 1 - \delta$, such i^* exists, and p_{i^*} is at least $(3/4 - 1/25)$, while for every i with $C_i \notin \mathsf{Comp}^{\Delta}_{1/3}(P)$, p_i is at most

$$(2/3 + 1/25) < (3/4 - 1/25) \leq p_{i^*},$$

which implies that the output of Sol is in $\mathsf{Comp}^{\Delta}_{1/3}(P)$. ◀

4.3 Proving Lemma 35

To prove Lemma 35, we start with the following claim.

▷ Claim 36. Let $P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell$ be a circuit, and assume that

$$\Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|\mathsf{Eve}(1^n, (r,k,\ell,P), x) - s| \leq \Delta(n)] \geq 9/10.$$

Then, on input P, Algorithm 34 outputs $C \in \mathsf{Comp}^{\Delta}_{1/4}(P)$ with probability at least $1/2$.

Proof of Claim 36. Let $P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell$ be a circuit such that

$$\Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|\mathsf{Eve}(1^n, (r,k,\ell,P), x) - s| \leq \Delta(n)] \geq 9/10.$$

Then, by definition it holds that

$$\mathrm{E}_{w\leftarrow\{0,1\}^{r_{\mathsf{Eve}}(n)}}\left[\Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|\mathsf{Eve}(1^n, (r,k,\ell,P), x) - s| > \Delta(n)]\right] \leq 1/10.$$

Using Markov's inequality, we gets that

$$\Pr_{w\leftarrow\{0,1\}^{r_{\mathsf{Eve}}(n)}}\left[\Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|\mathsf{Eve}(1^n, (r,k,\ell,P), x) - s| > \Delta(n)] > 1/4\right] \leq 1/2,$$

which implies that the circuit $C' = \mathsf{Eve}(1^n, (r,k,\ell,P), \cdot; w)$ is in $\mathsf{Comp}^{\Delta}_{1/4}(P)$ with probability at least $1/2$ over the choice of w, as we wanted to show. ◁

Given Claim 36, we are now ready to prove Lemma 35.

Proof of Lemma 35. Assume that Eve is such that

$$\Pr[|\mathsf{Eve}(1^n,(\mathbf{R}_n,\mathbf{K}_n,\mathbf{L}_n,\mathbf{P}_n),\mathbf{M}_n)-\mathbf{O}_n|\leq\Delta(1^n)]\geq 1-n^{-6}$$

for infinitely many n's. In the following, fix such large enough $n\in\mathbb{N}$. We show that

$$\Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|\mathsf{Eve}(1^n,(r,k,\ell,P),x)-s|\leq\Delta(n)]\geq 9/10 \tag{1}$$

for every $P\colon\{0,1\}^r\to\{0,1\}^k\times\{0,1\}^\ell$ with $P\in\mathcal{Q}_t^\Delta\cap\{0,1\}^n$. The proof then follows by Claim 36. To see the above, let $\mathbf{Pk}=(\mathbf{R}_n,\mathbf{K}_n,\mathbf{L}_n,\mathbf{P}_n))$, and notice that

$$\Pr_{\substack{\mathbf{R}_n,\mathbf{K}_n,\mathbf{L}_n,\mathbf{P}_n,\mathbf{C}_n\\ \mathbf{Pk}=(\mathbf{R}_n,\mathbf{K}_n,\mathbf{L}_n,\mathbf{P}_n)}}\left[\Pr_{\mathbf{M}_n,\mathbf{O}_n,\mathbf{W}\leftarrow\{0,1\}^r_{\mathsf{Eve}}}[|\mathsf{Eve}(1^n,\mathbf{Pk},\mathbf{M}_n;\mathbf{W})-\mathbf{O}_n|\leq\Delta(n)]<9/10\right] \tag{2}$$

$$\leq 10n^{-6}.$$

Indeed, it holds that

$$\begin{aligned}
1-n^{-6}&\leq\Pr[|\mathsf{Eve}(1^n,\mathbf{Pk},\mathbf{M}_n)-\mathbf{O}_n|\leq\Delta(n)]\\
&\leq\Pr_{\mathbf{Pk}}[\Pr[|\mathsf{Eve}(1^n,\mathbf{Pk},\mathbf{M}_n)-\mathbf{O}_n|\leq\Delta(n)]\geq 9/10]\\
&\qquad+9/10\cdot\Pr_{\mathbf{Pk}}[\Pr[|\mathsf{Eve}(1^n,\mathbf{Pk},\mathbf{M}_n)-\mathbf{O}_n|\leq\Delta(n)]<9/10]\\
&=(1-\Pr_{\mathbf{Pk}}[\Pr[|\mathsf{Eve}(1^n,\mathbf{Pk},\mathbf{M}_n)-\mathbf{O}_n|\leq\Delta(n)]<9/10])\\
&\qquad+9/10\cdot\Pr_{\mathbf{Pk}}[\Pr[|\mathsf{Eve}(1^n,\mathbf{Pk},\mathbf{M}_n)-\mathbf{O}_n|\leq\Delta(n)]<9/10]\\
&=1-1/10\cdot\Pr_{\mathbf{Pk}}[\Pr[|\mathsf{Eve}(1^n,\mathbf{Pk},\mathbf{M}_n)-\mathbf{O}_n|\leq\Delta(n)]<9/10]
\end{aligned}$$

which implies that Equation (2) holds. Next, we use Equation (2) and the upper bound on the computational depth of instances in $\mathcal{Q}_t^\Delta$, to show that Equation (1) holds for every $P\in\mathcal{Q}_t^\Delta$. To do so, fix $P\in\mathcal{Q}_t^\Delta\cap\{0,1\}^n$ and let $C\in\{0,1\}^n\cap\mathsf{Comp}_{n^{-10}}^{\Delta/n^{10}}(P)$ with $cd^{t,\infty}(C,P)\leq 2\log n$ be the circuit promised by the definition of $\mathcal{Q}_t^\Delta$. Assume towards a contradiction that

$$\Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|\mathsf{Eve}(1^n,(r,k,\ell,P),x)-s|\leq\Delta(n)]<9/10.$$

We want to upperbound $\mathsf{K}(P,C)$, to get a lower bound on the computational depth of (P,C). To this end, let $\mathcal{S}$ be the set of all pairs $(P',C')\in\{0,1\}^n\times\{0,1\}^n$, such that $C'\in\mathsf{Comp}_{n^{-10}}^{\Delta/n^{10}}(P')$, $\mathsf{K}^t(P',C')=\mathsf{K}^t(P,C)$, and on which Eve fails to approximate s with probability more than $1/10$.

By our assumption on (P,C), it holds that $(P,C)\in\mathcal{S}$. We next bound the size of $\mathcal{S}$. First, we claim that for every $(P',C')\in\mathcal{S}$,

$$\Pr[\mathbf{P}_n=P',\mathbf{C}_n=C']\geq 1/6n\cdot 2^{-\mathsf{K}^t(C,P)}. \tag{3}$$

Indeed, by definition there exists a program Π of length $\mathsf{K}^t(C,P)$ that outputs (C',P') in at most $t(2n)$ steps. Thus, Gen samples Π with probability at least $1/3n\cdot 2^{-\mathsf{K}^t(C,P)}$. Next, by Claim 32, the test in Step 3 of Gen passes with probability at least $1-2^{-n}>1/2$, since by definition of $\mathsf{Comp}_{n^{-10}}^{\Delta/n^{10}}$, $\Pr_{(x,s)\leftarrow P'(\mathbf{U}_r}\left[|C'(x)-s|\leq\Delta(1^n)/n^{10}\right]-n^{-9}\geq 1-n^{-10}-n^{-9}\geq 1-2n^{-7}$. In this case that the test passes, P' and C' are the output of Gen, and thus Equation (3) holds.

By Equation (3) and the definition of $\mathcal{S}$, we get that

$$\Pr_{\substack{\mathbf{R}_n,\mathbf{K}_n,\mathbf{L}_n,\mathbf{P}_n,\mathbf{C}_n \\ \mathbf{Pk}=(\mathbf{R}_n,\mathbf{K}_n,\mathbf{L}_n,\mathbf{P}_n)}}\left[\Pr_{\mathbf{M}_n,\mathbf{O}_n,\mathbf{W}\leftarrow\{0,1\}^r_{\mathsf{Eve}}}[|\mathsf{Eve}(1^n,\mathbf{Pk},\mathbf{M}_n;\mathbf{W})-\mathbf{O}_n|\leq\Delta(n)]<9/10\right]$$
$$\geq |\mathcal{S}|\cdot 1/6n\cdot 2^{-\mathsf{K}^t(C,P)}.$$

Combining the above with Equation (2) yields that

$$|\mathcal{S}|\leq\frac{10n^{-6}}{1/6n\cdot 2^{-\mathsf{K}^t(C,P)}}\leq 2^{\mathsf{K}^t(C,P)+\log n-6\log n+6}.$$

Observe that $\mathcal{S}$ can be (inefficiently) computed given $n,\mathsf{K}^t(C,P)$ and Eve. Thus, to encode (P,C), it is enough to encode $\mathcal{S}$ and the index of (P,C) in $\mathcal{S}$ (according to the lexicographic order). We conclude that,

$$\begin{aligned}\mathsf{K}(P,C)&\leq\mathsf{K}(n,\lambda,\mathsf{Eve})+2\log(\mathsf{K}(n,\lambda,\mathsf{Eve}))+\log|\mathcal{S}|+O(1)\\&\leq 2\log n+4\log\log n+\mathsf{K}^t(C,P)-5\log n+O(1)\\&<\mathsf{K}^t(C,P)-2\log n,\end{aligned}$$

where the last inequality holds for every large enough n. By the above, $\mathsf{K}^t(C,P)-\mathsf{K}(P,C)>2\log n$, in contradiction to the choice of (P,C). This yields that Equation (1) holds for every $P\in\mathcal{Q}_t$, as we wanted to show. ◀

4.4 Proving Theorem 26

We are now ready to use Lemma 15 in order to prove Theorem 26.

Proof of Theorem 26. By Lemmas 31 and 33, $(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ is a $(n^{-7},1-n^{-6},n^{10})$-weak-PKE (for $d(1^n)=\Delta(1^n)/n^{10}$), and thus it is also a $(n^{-6.9}/2,1-10n^{-6.1},2n^{6.9})$-weak-PKE. Thus, by Lemma 15, $(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ can be amplified into a PKE. ◀

▶ Remark 37 (The non-uniform setting). A similar theorem can be proven when assuming that $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t}\notin\mathsf{ioP}/\mathrm{poly}$, and when the PKE is secure against non-uniform adversaries. In this case, we assume that Eve is a non-uniform algorithm that breaks the PKE protocol, and want to construct a non-uniform (randomized) algorithm that decides $\mathsf{WBLearn}|_{\mathcal{Q}_t}$.

The issue with the above proof is that we cannot simply use Eve to bound the Kolmogorov complexity of (C,P) as done in the proof of Lemma 33, as Eve does not have constant size. However, we can find Eve using a small Turing machine: Let M be the (inefficient) Turing machine that, given a constant c such that n^c is a bound on the size of Eve, and an input for Eve, first find the circuit E'_n of size at most n^c that maximize the advantage in predicting M given an encryption of M by Enc, and then execute E' on the input. Observe that M has prediction advantage at least as the advantage of Eve. The theorem now follows using the same proof, by replacing Eve in the proof of Lemma 33 with M, and replacing Eve in Algorithm 34 with $E'=\{E'_n\}_{n\in\mathbb{N}}$.

References

1 Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *stoc29*, pages 284–293, 1997. See also ECCC TR96-065. doi:10.1145/258533.258604.

2 Michael Alekhnovich. More on average case vs approximation complexity. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 298–307. IEEE, 2003. doi:10.1109/SFCS.2003.1238204.

3 Luis Antunes and Lance Fortnow. Worst-case running times for average-case algorithms. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 298–303. IEEE, 2009. doi:10.1109/CCC.2009.12.

4 Luis Antunes, Lance Fortnow, Dieter Van Melkebeek, and N Variyam Vinodchandran. Computational depth: concept and applications. *Theoretical Computer Science*, 354(3):391–404, 2006. doi:10.1016/J.TCS.2005.11.033.

5 Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 171–180, 2010. doi:10.1145/1806689.1806715.

6 Marshall Ball, Yanyi Liu, Noam Mazor, and Rafael Pass. Kolmogorov comes to cryptomania: On interactive kolmogorov complexity and key-agreement. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 458–483. IEEE, 2023. doi:10.1109/FOCS57990.2023.00034.

7 Avrim Blum, Merrick Furst, Michael Kearns, and Richard J Lipton. Cryptographic primitives based on hard learning problems. In *Annual International Cryptology Conference*, pages 278–291. Springer, 1993. doi:10.1007/3-540-48329-2_24.

8 Andrej Bogdanov, Miguel Cueto Noval, Charlotte Hoffmann, and Alon Rosen. Public-key encryption from homogeneous clwe. In *Theory of Cryptography: 20th International Conference, TCC 2022, Chicago, IL, USA, November 7–10, 2022, Proceedings, Part II*, pages 565–592. Springer, 2022. doi:10.1007/978-3-031-22365-5_20.

9 Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 575–584, 2013. doi:10.1145/2488608.2488680.

10 Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *Journal of the ACM*, 43(2):831–871, 2014.

11 Gregory J. Chaitin. On the simplicity and speed of programs for computing infinite sets of natural numbers. *J. ACM*, 16(3):407–422, 1969. doi:10.1145/321526.321530.

12 Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, pages 644–654, 1976. doi:10.1109/TIT.1976.1055638.

13 Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Annual International Cryptology Conference (CRYPTO)*, pages 10–18, 1984.

14 Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009. doi:10.1145/1536414.1536440.

15 Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing (STOC)*, pages 25–32, 1989. doi:10.1145/73007.73010.

16 Danny Harnik, Joe Kilian, Moni Naor, Omer Reingold, and Alon Rosen. On robust combiners for oblivious transfer and other primitives. In *Advances in Cryptology–EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005. Proceedings 24*, pages 96–113. Springer, 2005. doi:10.1007/11426639_6.

17 J. Hartmanis. Generalized kolmogorov complexity and the structure of feasible computations. In *24th Annual Symposium on Foundations of Computer Science (sfcs 1983)*, pages 439–445, 1983. doi:10.1109/SFCS.1983.21.

18 Shuichi Hirahara and Mikito Nanashima. Learning in pessiland via inductive inference. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 447–457. IEEE, 2023. doi:10.1109/FOCS57990.2023.00033.

19 Thomas Holenstein. *Strengthening key agreement using hard-core sets*. PhD thesis, ETH Zurich, 2006. doi:10.3929/ETHZ-A-005205852.

20 Russell Impagliazzo and Leonid A. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *focs31*, pages 812–821, 1990. doi:10.1109/FSCS.1990.89604.

21 Michael Kearns and Leslie Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)*, 41(1):67–95, 1994. doi:10.1145/174644.174647.

22 Ker-I Ko. On the notion of infinite pseudorandom sequences. *Theor. Comput. Sci.*, 48(3):9–33, 1986. doi:10.1016/0304-3975(86)90081-2.

23 A. N. Kolmogorov. Three approaches to the quantitative definition of information. *International Journal of Computer Mathematics*, 2(1-4):157–168, 1968.

24 Yanyi Liu, Noam Mazor, and Rafael Pass. On white-box learning and public-key encryption. Cryptology ePrint Archive, Paper 2024/1931, 2024. URL: https://eprint.iacr.org/2024/1931.

25 Yanyi Liu and Rafael Pass. On one-way functions and kolmogorov complexity. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1243–1254. IEEE, 2020. doi:10.1109/FOCS46700.2020.00118.

26 Yanyi Liu and Rafael Pass. On one-way functions and the worst-case hardness of time-bounded kolmogorov complexity. *Cryptology ePrint Archive*, 2023.

27 Robert J McEliece. A public-key cryptosystem based on algebraic. *Coding Thv*, 4244:114–116, 1978.

28 Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 333–342, 2009.

29 Michael O Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical report, Massachusetts Inst of Tech Cambridge Lab for Computer Science, 1979.

30 Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009. doi:10.1145/1568318.1568324.

31 Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978. doi:10.1145/359340.359342.

32 Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999. doi:10.1137/S0036144598347011.

33 Michael Sipser. A complexity theoretic approach to randomness. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 330–335, 1983. doi:10.1145/800061.808762.

34 R.J. Solomonoff. A formal theory of inductive inference. part i. *Information and Control*, 7(1):1–22, 1964. doi:10.1016/S0019-9958(64)90223-2.

35 Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984. doi:10.1145/1968.1972.

36 Andrew C. Yao. Theory and applications of trapdoor functions. In *Annual Symposium on Foundations of Computer Science (FOCS)*, pages 80–91, 1982. doi:10.1109/SFCS.1982.45.

Sublinear Metric Steiner Tree via Improved Bounds for Set Cover

Sepideh Mahabadi
Microsoft Research, Redmond, WA, USA

Mohammad Roghani
Stanford University, CA, USA

Jakub Tarnawski
Microsoft Research, Redmond, WA, USA

Ali Vakilian
Toyota Technological Institute at Chicago (TTIC), Il, USA

Abstract

We study the metric Steiner tree problem in the sublinear query model. In this problem, for a set of n points V in a metric space given to us by means of query access to an $n \times n$ matrix w, and a set of terminals $T \subseteq V$, the goal is to find the minimum-weight subset of the edges that connects all the terminal vertices.

Recently, Chen, Khanna and Tan [SODA'23] gave an algorithm that uses $\widetilde{O}(n^{13/7})$ queries and outputs a $(2-\eta)$-estimate of the metric Steiner tree weight, where $\eta > 0$ is a universal constant. A key component in their algorithm is a sublinear algorithm for a particular set cover problem where, given a set system $(\mathcal{U}, \mathcal{F})$, the goal is to provide a multiplicative-additive estimate for $|\mathcal{U}| - \mathsf{SC}(\mathcal{U}, \mathcal{F})$. Here $\mathcal{U}$ is the set of elements, $\mathcal{F}$ is the collection of sets, and $\mathsf{SC}(\mathcal{U}, \mathcal{F})$ denotes the optimal set cover size of $(\mathcal{U}, \mathcal{F})$. In particular, their algorithm returns a $(1/4, \varepsilon \cdot |\mathcal{U}|)$-multiplicative-additive estimate for this set cover problem using $\widetilde{O}(|\mathcal{F}|^{7/4})$ membership oracle queries (querying whether a set $S \in \mathcal{S}$ contains an element $e \in \mathcal{U}$), where ε is a fixed constant.

In this work, we improve the query complexity of $(2-\eta)$-estimating the metric Steiner tree weight to $\widetilde{O}(n^{5/3})$ by showing a $(1/2, \varepsilon \cdot |\mathcal{U}|)$-estimate for the above set cover problem using $\widetilde{O}(|\mathcal{F}|^{5/3})$ membership queries. To design our set cover algorithm, we estimate the size of a random greedy maximal matching for an auxiliary multigraph that the algorithm constructs implicitly, without access to its adjacency list or matrix. Previous analyses of random greedy maximal matching have focused on simple graphs, assuming access to their adjacency list or matrix. To address this, we extend the analysis of Behnezhad [FOCS'21] of random greedy maximal matching on simple graphs to multigraphs, and prove additional properties that may be of independent interest.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Sublinear Algorithms, Steiner Tree, Set Cover, Maximum Matching, Approximation Algorithm

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.74

Related Version *Full Version*: https://arxiv.org/abs/2411.09059

Acknowledgements This work was done while Mohammad Roghani was an intern at Microsoft Research. The work was conducted in part while Sepideh Mahabadi and Ali Vakilian were long-term visitors at the Simons Institute for the Theory of Computing as part of the Sublinear Algorithms program.



© Sepideh Mahabadi, Mohammad Roghani, Jakub Tarnawski, and Ali Vakilian;
licensed under Creative Commons License CC-BY 4.0



1 Introduction

In the Steiner tree problem, we are given an undirected graph $G = (V, E)$, where each edge e has an associated cost $w(e)$, and a specified set of terminal vertices $T \subseteq V$. Then, the objective is to find a minimum-cost subgraph H of G that connects all terminals in T. The Steiner tree problem is one of the most fundamental problems in combinatorial optimization and has been extensively studied by the TCS community since it was included among Karp's 21 NP-Complete problems [34]. The state-of-the-art approximation factor for the Steiner tree problem is $\ln 4 + \varepsilon < 1.39$ [14], and it is known that approximating it to a factor better than 96/95 is NP-hard [17]. This Steiner tree problem has been studied in various domains, including approximation algorithms [40, 14], online algorithms [29, 3, 37, 26], stochastic algorithms [27, 25, 22], and massive data analysis models [15, 18, 16].

▶ **Definition 1** (Sublinear Metric Steiner Tree). *In the metric Steiner tree problem, we are given a set of points V, a set of terminal points $T \subseteq V$, and query access to an oracle $\mathcal{O}$ to the $|V| \times |V|$ distance matrix of a metric space (V, w), where $\mathcal{O}(u, v)$ returns the weight $w(u, v)$ of the edge (u, v).*

Let $\mathsf{ST}(V, T, w)$ denote the weight of a minimum-weight Steiner tree on instance (V, T, w). Then, the goal is to design an algorithm that estimates $\mathsf{ST}(V, T, w)$ using the fewest possible queries to the distance matrix via the oracle $\mathcal{O}$.

Czumaj and Sohler [18] presented the first sublinear query algorithm for the metric Steiner tree problem, showing a $(2 + \varepsilon)$-approximation using $\tilde{O}(k/\varepsilon^{O(1)})$ queries through their improved algorithm for the minimum spanning tree (MST) problem. Specifically, this follows their sublinear $(1 + \varepsilon)$-approximation for MST together with the well-known result by Gilbert and Pollak [23] showing that an α-approximation for MST over the metric induced on the terminals T is a (2α)-approximation for the metric Steiner tree instance with T as the terminal set.

Recently, Chen, Khanna, and Tan [16] studied the design of sublinear algorithms with strictly better-than-2 approximation for the metric Steiner tree problem. On the lower bound side, they showed that for any $\varepsilon > 0$, estimating the Steiner tree cost to within a $(5/3 - \varepsilon)$-factor requires $\Omega(n^2)$ queries, even when the number of terminals $|T|$ is constant. Moreover, they showed that for any $\varepsilon > 0$, estimating the Steiner tree cost to within a $(2 - \varepsilon)$-factor requires $\Omega(n + |T|^{6/5})$ queries. Additionally, they proved that for any $0 < \varepsilon < 1/3$, any algorithm that outputs a $(2 - \varepsilon)$-approximate Steiner tree (not just its cost) requires $\Omega(n|T|)$ queries. On the upper bound side, they showed that it is possible to achieve a better-than-2 estimate of the Steiner tree cost in sublinear time: there exists an algorithm that, with high probability, computes a $(2 - \eta)$-approximation of the Steiner tree cost using $\widetilde{O}(n^{13/7})$ queries, where $\eta > 0$ is a universal constant. At the core of their sublinear algorithm for metric Steiner tree with improved approximation guarantee, they relate the problem of achieving a better-than-2 estimation for the Steiner tree to a variant of set cover problem with a different objective.

▶ **Definition 2** (Threshold Set Cover). *Given a universe of elements $\mathcal{U}$ and a collection $\mathcal{F}$ of subsets of $\mathcal{U}$, in the Threshold Set Cover problem the goal is to estimate $\mathsf{ThSC}(\mathcal{U}, \mathcal{F}) \coloneqq |\mathcal{U}| - \mathsf{SC}(\mathcal{U}, \mathcal{F})$, where $\mathsf{SC}(\mathcal{U}, \mathcal{F})$ denotes the size of an optimal set cover solution for $(\mathcal{U}, \mathcal{F})$, i.e., the minimal number of sets in $\mathcal{F}$ whose union equals $\mathcal{U}$.*

Following the notation of [16] and for simplicity, in our technical sections, we also refer to this problem as set cover.

Specifically, given access to the adjacency matrix of the graph representation of $(\mathcal{U}, \mathcal{F})$, where there is an edge between $e \in \mathcal{U}$ and $S \in \mathcal{F}$ if and only if $e \in S$, Chen, Khanna, and Tan [16] designed an algorithm that, for any constant $0 < \varepsilon < 1$, with high probability, outputs a *multiplicative-additive* $(1/4, \varepsilon|\mathcal{U}|)$-approximation for estimation of $\mathsf{ThSC}(\mathcal{U}, \mathcal{F})$ using $\widetilde{O}_\varepsilon(|\mathcal{F}|^{3/2} + |\mathcal{F}|^{3/4} \cdot |\mathcal{U}|)$ queries to the adjacency matrix (or, membership queries). An estimate SOL for Threshold Set Cover on $(\mathcal{U}, \mathcal{F})$ is a *multiplicative-additive* (γ_1, γ_2)-approximation, if $\gamma_1 \cdot \mathsf{ThSC}(\mathcal{U}, \mathcal{F}) - \gamma_2 \leq \mathrm{SOL} \leq \mathsf{ThSC}(\mathcal{U}, \mathcal{F})$.

More broadly, there has been a large body of work on solving set cover problems in the massive data models of computation over the past decade [41, 19, 28, 20, 2, 30, 1, 5, 31, 24]. In particular the work of [31, 24] consider the set cover problem in the sublinear query model. However their algorithms assumes that it has an access to the adjacency list model as opposed to the adjacency matrix model, and thus cannot be directly employed here.

1.1 Our Results

Our key contribution is an algorithm for Threshold Set Cover, offering improved approximation guarantees and query complexity, as detailed below:

▶ **Theorem 3** (Our Algorithm for Threshold Set Cover). *There exists an algorithm that, given a set system $(\mathcal{U}, \mathcal{F})$ with oracle access to its adjacency matrix (also known as membership queries), outputs a multiplicative-additive $(1/2, \varepsilon \cdot |\mathcal{U}|)$-approximation to Threshold Set Cover, in $\widetilde{O}(|\mathcal{F}|^{5/3})$ time, with high probability.*

Note that both the query complexity and the running time of the algorithm are bounded by $\widetilde{O}(|\mathcal{F}|^{5/3})$, improving upon the algorithm by Chen, Khanna, and Tan [16] for large values of $|\mathcal{U}|$, which uses $\widetilde{O}_\varepsilon(|\mathcal{F}|^{3/2} + |\mathcal{F}|^{3/4} \cdot |\mathcal{U}|)$ membership queries and provides a multiplicative-additive $(1/4, \varepsilon|\mathcal{U}|)$-approximation for the problem. Notably, when $|\mathcal{U}| = \omega(|\mathcal{F}|^{2/3})$, the algorithm becomes sublinear in $|\mathcal{U}| \cdot |\mathcal{F}|$, making it especially relevant for applications in the metric Steiner tree problem. More specifically, our new algorithm for Threshold Set Cover results in the following improved sublinear query algorithm for the metric Steiner tree problem, which we show in Section 5.

▶ **Theorem 4** (Sublinear Algorithm for Metric Steiner Tree). *There exists an algorithm that, given an instance of metric Steiner tree denoted by (V, T, w) with oracle access $\mathcal{O}$ to the distance matrix of (V, w), outputs a $(2 - \eta)$-estimate of $\mathsf{ST}(V, T, w)$ using $\widetilde{O}(n^{5/3})$ queries to $\mathcal{O}$, where $\eta > 0$ is a universal constant, with high probability.*

Notably, the query complexity of our algorithm improves upon the $\widetilde{O}(n^{13/7})$ query complexity of the algorithm of Chen, Khanna, and Tan [16].

For a detailed overview of our technical contribution, see Section 2.

2 Technical Overview

In this section, we provide a brief overview of the technical challenges involved in designing our algorithms. To design a sublinear time algorithm for the Steiner tree problem, we use the framework developed by Chen, Khanna, and Tan [16]. They demonstrated that breaking the 2-approximation barrier for the Steiner tree problem can be reduced to solving an instance of a set cover problem. We refer the reader to Section 4.1 of [16] for details on this reduction.

We denote the variant of the set cover problem as *Threshold Set Cover*. Given a collection of sets $\mathcal{F}$ over a universe of elements $\mathcal{U}$, we aim to estimate the value of $\mathsf{ThSC}(\mathcal{U}, \mathcal{F}) = |\mathcal{U}| - \mathsf{SC}(\mathcal{U}, \mathcal{F})$, where $\mathsf{SC}(\mathcal{U}, \mathcal{F})$ denotes the size of the optimal set cover of the given instance.

To achieve our goal of breaking the 2-approximation barrier for the Steiner tree problem, we need to estimate $\mathsf{ThSC}(\mathcal{U}, \mathcal{F})$ with a $(\gamma, \varepsilon \cdot |\mathcal{U}|)$ multiplicative-additive error, where γ must be a constant and ε is any (small enough) constant. For this problem, we only have access to a membership oracle of the instance, meaning we can query whether a particular element e is in a particular set S or not. Note that this type of access is generally considered more challenging compared to an adjacency list oracle, where the algorithm can access either the ith element of a set S, or the ith set containing an element e. The reason is that if an element is included in only a constant number of sets, the algorithm is required to spend $\Omega(|\mathcal{F}|)$ queries to find just one set that contains the element. Consequently, we cannot use the results from the literature on sublinear set cover [24, 31] because they all rely on an adjacency list access model.

We will now provide an informal, step-by-step description of our algorithm for Threshold Set Cover, highlighting its differences, innovations, and technical challenges in comparison to the algorithm of [16]. For simplicity, in this technical overview we assume that $|\mathcal{F}| = \widetilde{\Theta}(|\mathcal{U}|)$, since this represents the worst-case scenario for the Steiner tree problem. However, our formal proof does not depend on this assumption. We let $n = |\mathcal{F}|$.

First step: sparsification of the Threshold Set Cover instance. The goal of this step is to produce a new instance where each element and each set has a low degree – specifically, where each element is in only a few sets, and each set contains only a few elements. This step is standard in designing sublinear algorithms for the set cover problem for different access models, and a slightly different version of it is also used in the algorithm by [16]. Our slight modification of the sparsification step allows us to relax some constraints in the reduction from the Steiner tree problem to Threshold Set Cover, enabling us to achieve the same query complexity for both problems.

Let $x > 0$ be some constant that we optimize later. Consider a set $S \in \mathcal{F}$ and suppose we randomly sample $\widetilde{O}(n^{1-x})$ elements from the universe and query the membership of all the sampled elements in S. If the size of S is at least $\widetilde{\Omega}(n^x)$, we expect to see a large intersection. Conversely, if the size of S is much smaller, we expect to see a small intersection. If a large intersection exists, we can remove the set S and all its elements from the instance. Since this event occurs at most $\widetilde{O}(n^{1-x})$ times, we can account for the removed elements and sets using the additive error in our estimation. Similarly, we can show that all elements belonging to more than $\widetilde{\Omega}(n^x)$ sets can be covered by a random subcollection of sets of size $\widetilde{O}(n^{1-x})$. Therefore, without loss of generality, by spending $\widetilde{O}(n^{2-x})$ time, we can assume that each set contains at most $\widetilde{O}(n^x)$ elements, and each element is included in at most $\widetilde{O}(n^x)$ sets.

Second step: constructing an auxiliary graph H and estimating the size of its maximum matching. Similar to [16], we construct a graph H with a vertex set where each vertex corresponds to an element of $\mathcal{U}$. We connect two vertices if their corresponding elements appear together in at least one set from $\mathcal{F}$. It is important to note that we do not construct H explicitly, as doing so would be computationally expensive and require $\Omega(n^2)$ time. As shown by [16], if the size of the maximum matching of H is large, it is evident that $\mathsf{ThSC}(\mathcal{U}, \mathcal{F})$ is significantly smaller than $|\mathcal{U}|$. Conversely, if the size of the maximum matching of H is close to zero, then $\mathsf{ThSC}(\mathcal{U}, \mathcal{F})$ is also close to zero. This is sufficient for our purposes, as our goal is to obtain a constant-factor approximation. Intuitively, each matching edge in H indicates that there are two elements that can be covered together, which increases the value of $\mathsf{ThSC}(\mathcal{U}, \mathcal{F})$.

There is extensive literature on estimating the size of maximum matching in sublinear time [4, 6, 8, 7, 9, 12, 13, 33, 36, 38, 39, 43], with significant progress made in recent years. For our application, we use the algorithm of Behnezhad [6] to estimate the size of a random greedy maximal matching (RGMM) of the graph. In summary, this algorithm can estimate the size of the RGMM of a graph in $\widetilde{O}(\bar{d})$ time if given access to the adjacency list of the graph, where $\bar{d}$ denotes the average degree of the graph. We can now use this algorithm as a black box:

- The average degree of H is $\widetilde{O}(n^{2x})$, since each element is in $\widetilde{O}(n^x)$ sets and each set contains $\widetilde{O}(n^x)$ elements.
- Each time the algorithm visits a vertex in H (corresponding to an element), we can spend $\widetilde{O}(n^{1+x})$ time to find its adjacency list in H. This involves first querying all sets that include the element, and then making queries between those sets and all elements.

Therefore, we can simulate the algorithm from [6] in $\widetilde{O}(\bar{d} \cdot n^{1+x}) = \widetilde{O}(n^{1+3x})$ time. By balancing this with the sparsification step, which requires $\widetilde{O}(n^{2-x})$ time, we can set $x = 1/4$ to achieve an algorithm with a running time of $\widetilde{O}(n^{7/4})$. This is essentially the running time of the algorithm by Chen, Khanna, and Tan [16].

Third step: using the algorithm of Behnezhad [6] in a white-box manner. To improve the running time of our algorithm, we need to open up the RGMM algorithm from [6] and utilize its properties to apply it more effectively. The RGMM algorithm is a local algorithm that explores the neighborhood of a given vertex to determine whether it is matched. A key observation is that during each exploration, the algorithm requires a random neighbor of the vertex that has not been explored yet. However, in the previous approach, we constructed the entire adjacency list of the vertex, which is redundant and inefficient. Intuitively, we only need to randomly identify one of the vertex's neighbors in each step.

However, the first challenge we encounter is that we cannot select a neighbor uniformly at random. To illustrate this, consider the following example. Suppose that we have five elements $\mathcal{U} = \{e_1, \ldots, e_5\}$ and three sets: $S_1 = \{e_1, e_2, e_3\}$, $S_2 = \{e_1, e_2, e_4\}$, and $S_3 = \{e_1, e_2, e_5\}$. Suppose that we want to find a random neighbor of e_1 in H. If we first find all sets that include e_1 and then query between those sets and all elements uniformly at random until we find an edge in H, we are likely to see the edge (e_1, e_2) because it appears in all sets. Consequently, the algorithm has a bias towards finding neighbors that appear in more sets with the element.

To overcome this challenge, rather than defining an auxiliary simple graph H, we define an auxiliary multigraph H. In this multigraph, if two elements appear in the same set multiple times, we add an edge for each of those occurrences. Note that the average degree of H remains at most $\widetilde{O}(n^{2x})$. However, the algorithm and analysis for RGMM from [6] are designed for simple graphs. We extend these results to multigraphs and show that we can estimate the size of RGMM for a multigraph, given access to its adjacency list. This extension may be of independent interest and could be useful for tackling other problems in sublinear time. To establish this, we build on the exquisite approach first introduced by Yoshida, Yamamoto, and Ito [43] and further explored in various settings [6, 10, 11]. We employ techniques such as the analysis of the round-complexity of maximal independent sets [21], double-counting arguments to bound the average complexity of RGMM on multigraphs, and others; we encourage the reader to refer to the arXiv version of the paper for further details.

Now, suppose that for each vertex the RGMM algorithm explores in H, we first query all sets to identify those that include the corresponding element. Since the algorithm explores at most $\widetilde{O}(\bar{d})$ vertices in H, this step will cost at most $\widetilde{O}(\bar{d} \cdot n) = \widetilde{O}(n^{1+2x})$ in total. Let v be a vertex that the RGMM algorithm is exploring at the moment. Define $\mathcal{S}_v$ to be the collection

of sets that include element v. Now, if we query uniformly at random between all elements and the collection $\mathcal{S}_v$, each incident edge of v in the multigraph H has an equal probability of being sampled, which resolves the first challenge. For now, assume that the degree of all vertices in H is $\bar{d}$. Since $|S_v| = \widetilde{O}(n^x)$ and there are $\widetilde{O}(n)$ elements in total, we expect to find an element in one of the sets of S_v every $\widetilde{O}(n^{1+x}/\bar{d})$ queries. Thus, to identify a random neighbor of a vertex in H, we need to spend $\widetilde{O}(n^{1+x}/\bar{d})$ time. The RGMM algorithm queries for a random neighbor of a vertex $\widetilde{O}(\bar{d})$ times, since the exploration size is $\widetilde{O}(\bar{d})$; therefore, the total cost is $\widetilde{O}(n^{1+x})$. Combining this with the cost of sparsification, the total cost of the algorithm is $\widetilde{O}(\max(n^{2-x}, n^{1+2x}))$, which is $\widetilde{O}(n^{5/3})$ if we set $x = 1/3$.

The second challenge arises because the RGMM algorithm may predominantly visit vertices with a very low degree in H. For such vertices, finding a random neighbor can be much more time-consuming. Generally, if a vertex v in H has degree $\deg_H(v)$, then each time the algorithm finds a random neighbor of v, it needs to spend $\widetilde{O}(n^{1+x}/\deg_H(v))$ time. Therefore, if the algorithm frequently encounters vertices with constant degree, each query to find a random neighbor can cost $\widetilde{O}(n^{1+x})$. With the exploration size being $\widetilde{O}(\bar{d})$, this can significantly increase the query complexity of the algorithm. As a property of the local RGMM algorithm, we demonstrate that each vertex is visited by the algorithm in proportion to its degree in H. More formally, we prove that RGMM requires $\widetilde{O}(\deg_H(v)/n)$ neighbors of v on average, for a uniformly random permutation of edges. Thus, the degree-dependent factors cancel each other out, and the average cost of this part can be upper-bounded by $\widetilde{O}(n^{1+x})$, which is enough for us to get the $\widetilde{O}(n^{5/3})$ running time for the Threshold Set Cover.

3 Preliminaries

As is common in the literature, we use the term "with high probability" to refer to a probability of at least $1 - n^{-\alpha}$, for a sufficiently large constant $\alpha \geq 2$. Moreover, we use $\widetilde{O}(\cdot)$, $\widetilde{\Theta}(\cdot)$, and $\widetilde{\Omega}(\cdot)$ to hide the dependency on $\text{poly}(\log n)$. For a maximization problem of estimating some value χ, and for $\gamma_1 \in (0, 1]$ and $\gamma_2 > 0$, we say that $\widetilde{\chi}$ is a multiplicative-additive (γ_1, γ_2)-approximation of the value χ if $\gamma_1\chi - \gamma_2 \leq \widetilde{\chi} \leq \chi$.

3.1 Probabilistic Tools

We use the following standard concentration inequalities in our proof.

▶ **Proposition 5** (Chernoff Bound). *Let $X_1, X_2, \ldots, X_n$ be n independent Bernoulli random variables. Let $X = \sum_{i=1}^{n} X_i$. For any $k > 0$, it holds that*

$$\Pr[|X - \mathbf{E}[X]| \geq k] \leq 2\exp\left(-\frac{k^2}{3\,\mathbf{E}[X]}\right).$$

▶ **Definition 6** (Negative Association [32, 35, 42]). *Let $X_1, X_2, \ldots, X_n$ be a set of random variables. We say this set is negatively associated if for any two disjoint index sets $I, J \subseteq [n]$, and two functions f and g, both either monotonically increasing or monotonically decreasing, the following condition is satisfied:*

$$\mathbf{E}[f(X_i : i \in I) \cdot g(X_j : j \in J)] \leq \mathbf{E}[f(X_i : i \in I)] \cdot \mathbf{E}[g(X_j : j \in J)].$$

▶ **Proposition 7** (Chernoff Bound for Negatively Associated Variables). *Let $X_1, X_2, \ldots, X_n$ be a set of negatively associated Bernoulli random variables. Let $X = \sum_{i=1}^{n} X_i$. Then*

$$\Pr\left[X \geq (1+\alpha)\,\mathbf{E}[X]\right] \leq \left(\frac{e^{\alpha}}{(1+\alpha)^{1+\alpha}}\right)^{\mathbf{E}[X]}$$

and

$$\Pr\left[X \leq (1-\alpha)\,\mathbf{E}[X]\right] \leq \left(\frac{e^{-\alpha}}{(1-\alpha)^{1-\alpha}}\right)^{\mathbf{E}[X]}$$

▶ **Proposition 8** (Markov Inequality). *Let X be a non-negative random variable. For any $\alpha > 0$, it holds that*

$$\Pr[X \geq \alpha] \leq \frac{\mathbf{E}[X]}{\alpha}.$$

3.2 Graph Theory

A *multigraph* is a type of graph in which multiple edges, also known as parallel edges, are allowed between any pair of vertices. A *line graph* of a graph G is a graph that represents the adjacencies between the edges of G. More formally, given a graph G, the line graph $L(G)$ is constructed as follows:

- **Vertices**: Each vertex in $L(G)$ corresponds to an edge in G.
- **Edges**: Two vertices in $L(G)$ are connected by an edge if and only if their corresponding edges in G share a common endpoint (i.e., they are incident to the same vertex in G).

For a graph G, we let $\deg_G(v)$ be the degree of vertex v. In the case of multigraphs, $\deg_G(v)$ counts parallel edges multiple times.

Random Greedy Maximal Matching (RGMM). Given a graph $G = (V, E)$, a random greedy maximal matching is constructed by first selecting a random permutation π of the edges E. The algorithm then iterates over the edges in the order specified by π, adding each edge to the matching if neither of its endpoints is already matched (i.e., if none of its adjacent edges have been included in the matching so far). This process continues until all edges have been considered. The term "random" comes from the fact that the permutation π is chosen uniformly at random among all possible permutations of the edges.

Random Greedy Maximal Independent Set (MIS). Given a graph $G = (V, E)$, a random greedy maximal independent set is constructed by first selecting a random permutation π of the vertices V. The algorithm then iterates over the vertices in the order specified by π, adding each vertex to the independent set if none of its neighbors are already in the set (i.e., it does not share an edge with any vertex already included in the independent set). This process continues until all vertices have been considered. The term "random" comes from the fact that the permutation π is chosen uniformly at random among all possible permutations of the vertices.

Parallel Randomized Greedy Maximal Independent Set. Let G be a graph, and let π be a permutation of its vertices. In each iteration, we select all vertices whose rank is lower than that of all their neighbors and then remove these vertices along with their neighbors from the graph. The number of iterations required for G to become empty is referred to as the round complexity, denoted by $\rho(G, \pi)$. It is not hard to see that the MIS produced by the parallel randomized greedy maximal independent set is the same as the output of the random greedy maximal independent set for a fixed permutation π.

4 Sublinear Algorithm for Set Cover

In this section, we formalize and analyze our algorithm for the set cover variant described above. Throughout this section, we assume that $\mathcal{U}$ denotes the universe and $\mathcal{F}$ denotes the collection of sets. We will slightly abuse notation by letting $k = |\mathcal{U}|$ and $n = |\mathcal{F}|$. Without loss of generality, we can assume that $n \geq k$.[1] We use $\mathsf{SC}(\mathcal{U}, \mathcal{F})$ for the size of the minimum set cover of the input instance. Our goal is to design an algorithm that estimates the value of $\chi = k - \mathsf{SC}(\mathcal{U}, \mathcal{F})$ with at most εk additive error and a constant multiplicative factor. For $\gamma_1 \in (0, 1]$ and $\gamma_2 > 0$, we say that $\widetilde{\chi}$ is a multiplicative-additive (γ_1, γ_2)-approximation of the value χ if $\gamma_1 \chi - \gamma_2 \leq \widetilde{\chi} \leq \chi$. Similar to the reduction from the Steiner tree problem to set cover in [16], we need to estimate $k - \mathsf{SC}(\mathcal{U}, \mathcal{F}_{\neq 2})$ where $\mathcal{F}_{\neq 2}$ denotes the collection of all sets in $\mathcal{F}$ except those of size exactly 2. For simplicity, we focus on estimating χ, and in the final step of this section, we will explain how to handle sets of size 2 in our algorithm. For our application, we need γ_1 to be constant and $\gamma_2 = \varepsilon k$ where ε is a small fixed constant.

A High-Level Description of the Algorithm. Our algorithm for estimating the value of χ is formalized in Algorithm 1. Apart from the collection of sets $\mathcal{F}$ and the universe of elements $\mathcal{U}$, the algorithm runs with two parameters, α and β, both of which can be determined based on the values of x and y, which we optimize in the final step of the analysis. The algorithm has three phases: 1) sparsification of the sets, 2) sparsification of the elements, and 3) estimating the size of a maximum matching of the auxiliary graph H (Definition 9). In the following, we first describe each component in words before moving on to the formal proofs.

(Step 1) Set Sparsification. This component of the algorithm is formalized in Algorithm 2. The algorithm maintains a collection of sets $\hat{\mathcal{F}}$ and a universe of elements $\hat{\mathcal{U}}$ that are initially equal to $\mathcal{F}$ and $\mathcal{U}$, respectively. We iterate over all sets in $\mathcal{F}$ one by one and for each set S, we sample $r_1 = |\hat{\mathcal{U}}|/\alpha$ random elements from $\hat{\mathcal{U}}$. Intuitively, if $|S \cap \hat{\mathcal{U}}| \geq \widetilde{\Omega}(\alpha)$, we expect to have an element of S in the r_1 random sampled elements. Having this in mind, if there is a large enough intersection ($\Omega(\log n)$) between the sampled elements and S, we remove set S and all its elements from $\hat{\mathcal{F}}$ and $\hat{\mathcal{U}}$, respectively (we add this set to our solution). Therefore, after the execution of the algorithm, each remaining set in $\hat{\mathcal{F}}$ has at most $\widetilde{O}(\alpha)$ elements. On the other hand, if $|S \cap \hat{\mathcal{U}}|$ is smaller than α, we expect to see a small intersection with the r_1 sampled elements. Consequently, the number of times the algorithm removes a set and its elements from $\hat{\mathcal{F}}$ and $\hat{\mathcal{U}}$ is at most $k/\alpha = o(k)$, which can be accounted for by the additive error in the estimation. Also, if at any point the size of the maintained universe becomes smaller than some threshold (in the algorithm the value of the threshold is $\widetilde{\Theta}(\alpha)$, the algorithm stops processing the rest of the sets (Algorithm 2) since $\hat{\mathcal{U}} = \widetilde{O}(\alpha)$. This step differs from the algorithm in [16] because we sequentially sparsify the sets, whereas their approach is non-adaptive.

(Step 2) Sparsification of Elements. This part of the algorithm is formalized in Algorithm 3. Similar to the previous step, we want to sparsify our instance such that each element in the remaining instance appears in at most $\widetilde{O}(\beta)$ sets. Let $\hat{\mathcal{U}}$ and $\hat{\mathcal{F}}$ be the output of Algorithm 2. We sample $r_2 = |\hat{\mathcal{U}}|/\beta$ random sets from the collection $\hat{\mathcal{F}}$. With the same intuition as the

[1] For the sake of this problem, for each element in the universe we can add a set that only contains the element. The same assumption is also made in [16].

previous step, if some element is in at least $\widetilde{\Omega}(\beta)$ sets of $\hat{\mathcal{F}}$, we expect to see it in many sampled sets. We partition the elements of $\hat{\mathcal{U}}$ into $\mathcal{U}_{low}$ and $\mathcal{U}_{high}$, depending on whether their intersection with the randomly sampled elements is smaller than a given threshold or not. With high probability, each element in $\mathcal{U}_{low}$ appears in at most $\widetilde{O}(\beta)$ sets of $\hat{\mathcal{F}}$, and each element in $\mathcal{U}_{high}$ appears in at least $\widetilde{\Omega}(\beta)$ sets of $\hat{\mathcal{F}}$. This suffices to show that any random subset of $\hat{\mathcal{F}}$ of size $\varepsilon k/2$ can cover all elements of $\mathcal{U}_{high}$, which can be included in the additive error of the estimation. This step is similar to the approach used in [16].

After steps 1 and 2 of the algorithm, we have the property that each set in the remaining instance has at most $\widetilde{O}(\alpha)$ elements, and each element in the remaining instance is in at most $\widetilde{O}(\beta)$ sets.

(Step 3) Estimating the Maximum Matching of Auxiliary Graph H. We construct an auxiliary multigraph H with vertex set $\mathcal{U}_{low}$. For each set $S \in \hat{\mathcal{F}}$, we add an edge in H between every two elements of S. Note that we do not explicitly construct the multigraph H because doing so would require $\Omega(nk)$ time, which is not feasible. We now estimate the size of the maximum matching in H to produce our final estimate. Intuitively, if χ is very small (close to zero), meaning that nearly k sets are required to cover the universe, then the maximum matching in H will also be small. To see this, note that a matching edge implies that its two endpoints can be covered by the same set. Conversely, if χ is large (almost equal to k), then H will have a large matching because each set in the set cover solution covers many new elements, which can be almost paired up in H. Thus, obtaining a constant approximation for the size of the maximum matching in H is sufficient to achieve our goal, and we do this by estimating the size of a random greedy maximal matching in H. To that end, we modify and analyze the algorithm from [6] for multigraphs and adapt it to our access model for the multigraph H, as discussed in detail in the full version of the paper. The algorithm in [16] constructs a similar graph, but it is not a multigraph in the sense that, for any two elements that appear together in multiple sets, only a single edge is added between them in H.

▶ **Definition 9** (Auxiliary Multigraph H). *Let $\hat{\mathcal{F}}$ and $\mathcal{U}_{low}$ be as defined in Algorithm 1, respectively, of Algorithm 1. We construct an auxiliary graph H with vertex set $\mathcal{U}_{low}$ such that for each set S in $\hat{\mathcal{F}}$ and each two different elements $e, e' \in \mathcal{U}_{low}$, we add an edge (e, e') to H. Note that H is a multigraph: multiple sets may contain both elements e and e', and we add an edge for each of these sets.*

Algorithm 1 Sublinear Time Algorithm for Set Cover.

Input: Collection of sets $\mathcal{F}$ and universe of elements $\mathcal{U}$.
Parameter: $\alpha \leftarrow n^x, \quad \beta \leftarrow 10 \max(k/n^{1-y}, 1) \cdot n \log(n)/k$.
Let $\hat{\mathcal{F}}$ and $\hat{\mathcal{U}}$ be the output of Algorithm 2 with input $\mathcal{F}$, $\mathcal{U}$ and α.
Let $\mathcal{U}_{low}$ and $\mathcal{U}_{high}$ be the output of Algorithm 3 with input $\hat{\mathcal{F}}$, $\hat{\mathcal{U}}$, and β.
Let H be the auxiliary multigraph defined in Definition 9. ▷ We do not build H explicitly.
Let $\widetilde{\mu}$ be the estimate of $\mathbf{E}_\pi |\mathrm{RGMM}(H, \pi)|$.
Let $\widetilde{\chi} \leftarrow \widetilde{\mu} + |\mathcal{U} \setminus \mathcal{U}_{low}| - \varepsilon k/2$.
return $\widetilde{\chi}$.

Algorithm 2 Sparsification of Sets.

Input: Collection of sets $\mathcal{F}$, universe of elements $\mathcal{U}$, and parameter α that controls the sparsification ratio.
$\hat{\mathcal{F}} \leftarrow \mathcal{F}, \quad \hat{\mathcal{U}} \leftarrow \mathcal{U}, \quad c \leftarrow 0.$ ▷ We use c only for the analysis.
for $S \in \mathcal{F}$ **do**
  if $|\hat{\mathcal{U}}| < 10\alpha \log n$ **then**
    break
  $r_1 \leftarrow |\hat{\mathcal{U}}|/\alpha.$
  Let $e_1, e_2, \ldots, e_{r_1}$ be r_1 random elements of $\hat{\mathcal{U}}$.
  Make queries between S and elements $e_1, \ldots, e_{r_1}$.
  if $|\{e_1, \ldots, e_{r_1}\} \cap S| \geq 10 \log n$ **then**
    $\hat{\mathcal{F}} \leftarrow \hat{\mathcal{F}} \setminus \{S\}.$
    $c \leftarrow c + 1.$
    for $e \in \hat{\mathcal{U}}$ **do**
      if $e \in S$ **then**
        $\hat{\mathcal{U}} \leftarrow \hat{\mathcal{U}} \setminus \{e\}.$
return $\hat{\mathcal{F}}, \hat{\mathcal{U}}$

Algorithm 3 Sparsification of Elements.

Input: Collection of sets $\hat{\mathcal{F}}$, universe of elements $\hat{\mathcal{U}}$, and parameter β that controls the sparsification ratio.
$r_2 \leftarrow |\hat{\mathcal{U}}|/\beta.$
if $r_2 < 20 \log n/\varepsilon$ **then**
  $\mathcal{U}_{low} \leftarrow \hat{\mathcal{U}}, \quad \mathcal{U}_{high} \leftarrow \emptyset.$
  return $\mathcal{U}_{low}, \mathcal{U}_{high}.$
Let $\{S_1, S_2, \ldots, S_{r_2}\}$ be r_2 random sets from $\hat{\mathcal{F}}$.
Make queries between all elements in $\hat{\mathcal{U}}$ and sets in $\{S_1, S_2, \ldots, S_{r_2}\}$.
Let $\mathcal{U}_{low}$ be the elements that appeared in at most $20 \log n/\varepsilon$ many sets.
$\mathcal{U}_{high} \leftarrow \hat{\mathcal{U}} \setminus \mathcal{U}_{low}.$
return $\mathcal{U}_{low}, \mathcal{U}_{high}.$

4.1 Proof of Correctness

In this section, we prove the correctness of Algorithm 1.

▷ Claim 10. For any set S such that the condition of Algorithm 2 holds during the execution of Algorithm 2, with high probability it holds that $|S \cap \hat{\mathcal{U}}| \geq \alpha$, where $\hat{\mathcal{U}}$ denotes the universe that Algorithm 2 maintains at the time it processes S.

Proof. Suppose that S is a set such that $|S \cap \hat{\mathcal{U}}| < \alpha$ at the time that Algorithm 2 processes this set. Let X_i be the random variable that indicates $e_i \in S$. Thus, we have $\Pr[X_i] \leq |S \cap \hat{\mathcal{U}}|/|\hat{\mathcal{U}}|$. Let $X = \sum_{i=1}^{r_1} X_i$. By linearity of expectation, we have $\mathbf{E}[X] < r_1\alpha/|\hat{\mathcal{U}}| = 1$. Also, note that X_i's are negatively associated random variables. Let $\lambda = (9 \log n)/\mathbf{E}[X]$. Therefore, using the Chernoff bound for negatively associated random variables (Proposition 7) we have

$$\begin{aligned}\Pr[X \geq (1+\lambda)\,\mathbf{E}[X]] &\leq \left(\frac{e^{\lambda}}{(1+\lambda)^{1+\lambda}}\right)^{\mathbf{E}[X]} \\ &\leq \left(\frac{e^{\lambda}}{\lambda^{\lambda}}\right)^{\mathbf{E}[X]} && \text{(Since } \lambda > 1\text{)} \\ &= \left(\frac{e}{\lambda}\right)^{9\log n} && \text{(Since } \lambda = (9\log n)/\,\mathbf{E}[X]\text{)} \\ &\leq \frac{1}{n^9} && \text{(Since } \lambda > e^2\text{)}\end{aligned}$$

which implies that with probability of at least $1 - n^{-9}$,

$$X < (1+\lambda)\,\mathbf{E}[X] = \mathbf{E}[X] + 9\log n < 10\log n.$$

Since we have n sets, using a union bound, with a probability at least $1 - n^{-8}$, for any set such that the condition of Algorithm 2 holds, we have $|S \cap \hat{\mathcal{U}}| \geq \alpha$. ◁

▷ Claim 11. Let c be the variable used in Algorithm 2. With high probability, we have $c \leq k/\alpha$.

Proof. By Claim 10, for every set S such that the condition on Algorithm 2 of Algorithm 2 holds, with high probability we have that $|S \cap \hat{\mathcal{U}}| \geq \alpha$. Hence, each time the algorithm increases c, the size of $\hat{\mathcal{U}}$ decreases by α. Therefore, the total number of times the algorithm increases c is upper-bounded by k/α. ◁

▷ Claim 12. Let $\hat{\mathcal{F}}$ and $\hat{\mathcal{U}}$ be the output of Algorithm 2. Then, each set $S \in \hat{\mathcal{F}}$ has at most $20\alpha\log n$ elements in $\hat{\mathcal{U}}$ with high probability.

Proof. First, note that if the algorithm stops because of the condition of Algorithm 2 and does not process S, it holds that $|\hat{\mathcal{U}}| < 10\alpha\log n$ and the claim trivially holds.

Let S be a set such that at the time that Algorithm 2 processes S, we have $|S \cap \hat{\mathcal{U}}| \geq 20\alpha\log n$. Similar to the proof of Claim 10, let X_i be the random variable that indicates $e_i \in S$ and $X = \sum_{i=1}^{r_1} X_i$. Hence, $\mathbf{E}[X] \geq 20 r_1 \alpha \log n/|\hat{\mathcal{U}}| = 20\log n$. Since X_i's are negatively associated random variables, using Chernoff bound (Proposition 7) for $\lambda = (9\log n)/\,\mathbf{E}[X]$, we have

$$\begin{aligned}\Pr[X \leq (1-\lambda)\,\mathbf{E}[X]] &\leq \left(\frac{e^{\lambda}}{(1+\lambda)^{1+\lambda}}\right)^{\mathbf{E}[X]} \\ &\leq \left(\frac{e^{\lambda}}{\lambda^{\lambda}}\right)^{\mathbf{E}[X]} && \text{(Since } \lambda > 1\text{)} \\ &= \left(\frac{e}{\lambda}\right)^{9\log n} && \text{(Since } \lambda = (9\log n)/\,\mathbf{E}[X]\text{)} \\ &\leq \frac{1}{n^9} && \text{(Since } \lambda > e^2\text{)}\end{aligned}$$

which implies that with a probability of at least $1 - n^{-9}$,

$$X > (1-\lambda)\,\mathbf{E}[X] = \mathbf{E}[X] - 9\log n \geq 20\log n - 9\log n > 10\log n.$$

Therefore, the condition on Algorithm 2 of Algorithm 2 must hold for all such sets with a probability of at least $1 - n^{-8}$ using union bound. ◁

▶ **Lemma 13** (Sets Sparsification Guarantee). *Let $\hat{\mathcal{F}}$ and $\mathcal{U}_{low}$ be outputs of Algorithm 2 and Algorithm 3. Also, let $S \in \hat{\mathcal{F}}$. Then, with high probability, S contains at most $\widetilde{O}(\alpha)$ elements of $\mathcal{U}_{low}$.*

Proof. Note that $\mathcal{U}_{low} \subseteq \hat{\mathcal{U}}$ where $\hat{\mathcal{U}}$ is the output of Algorithm 2. Also, by Claim 12, we have $|S \cap \hat{\mathcal{U}}| \leq 20\alpha \log n$. Thus, with high probability we have $|S \cap \mathcal{U}_{low}| \leq 20\alpha \log n = \widetilde{O}(\alpha)$. ◀

▶ **Lemma 14** (Elements Sparsification Guarantee). *Let $\mathcal{U}_{low}$ be as output by Algorithm 3 and let $e \in \mathcal{U}_{low}$. Then, with high probability, there are at most $\widetilde{O}(\beta)$ sets of $\hat{\mathcal{F}}$ that contain e.*

Proof. Let e be an element of $\hat{\mathcal{U}}$, where $\hat{\mathcal{U}}$ is the output of Algorithm 2. We show that if at least $40\beta \log n/\varepsilon$ sets in $\hat{\mathcal{F}}$ contain e, then $e \in \mathcal{U}_{high}$ in the output of Algorithm 3 with high probability.

Let X_i be an indicator variable for S_i containing e. Thus, we have $\mathbf{E}[X_i] \geq 40\beta \log n/(\varepsilon n)$. Define $X = \sum_{i=1}^{r_2} X_i$. Hence, $\mathbf{E}[X] \geq r_2 \cdot 40\beta \log n/(\varepsilon n) = 40 \log n/\varepsilon$. Since X_i's are negatively associated random variables, using Chernoff bound (Proposition 7) for $\lambda = (9 \log n)/\mathbf{E}[X]$, we have

$$\Pr[X \leq (1-\lambda)\mathbf{E}[X]] \leq \frac{1}{n^9} \qquad \text{(Similar to the proof of Claim 12)}$$

which implies that $X > 20 \log n$ with a probability of at most $1 - n^{-9}$, since

$$X > (1-\lambda)\mathbf{E}[X] = \mathbf{E}[X] - 9\log n \geq 40 \log n/\varepsilon - 9 \log n > 20 \log n/\varepsilon.$$

Using a union bound for all elements in $\hat{\mathcal{U}}$ that are in at least $40\beta \log n/\varepsilon$ sets of $\hat{\mathcal{F}}$, with high probability all of them are going to be included in $\mathcal{U}_{high}$. As a result, for each $e \in \mathcal{U}_{low}$, e is in at most $\widetilde{O}(\beta)$ sets of $\hat{\mathcal{F}}$. ◀

▷ Claim 15. Let $\mathcal{F}'$ be a random collection of $\varepsilon k/5$ sets of $\hat{\mathcal{F}}$. Then, with high probability, every element in $\mathcal{U}_{high}$ is in one of the sets of $\mathcal{F}'$.

Proof. Let $e \in \hat{\mathcal{U}}$ be such that at most $15\beta \log n/\varepsilon$ sets of $\hat{\mathcal{F}}$ contain e. Similar to the proof of Lemma 14, let X be a random variable that denotes the number of sets S_i (for $i = 1, 2, ..., r_2$) that contain e. Using a Chernoff bound, we can prove that with a probability of at least $1 - n^{-2}$, for all such elements e, we have $X < 20 \log n/\varepsilon$. Therefore, all elements in $\mathcal{U}_{high}$ are in at least $15\beta \log n/\varepsilon$ sets of $\hat{\mathcal{F}}$.

Consequently, when $|\mathcal{F}'| \geq \varepsilon k/5$, the expected number of sets in $\mathcal{F}'$ that cover element $e \in \mathcal{U}_{high}$ is at least

$$\frac{15\beta \log n}{\varepsilon} \cdot \frac{1}{n} \cdot \frac{\varepsilon k}{5} \geq \frac{15n \log n}{\varepsilon k} \cdot \frac{1}{n} \cdot \frac{\varepsilon k}{5} = 3 \log n$$

where we used that $\beta \geq n/k$ (see Algorithm 1 in Algorithm 1). Using Chernoff bounds again, we expect all these elements to be covered by at least one set with high probability, which concludes the proof. ◁

▷ Claim 16. Let $\mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}})$ be the optimal set cover size for the universe of elements $\mathcal{U}_{low}$ and the collection of sets $\hat{\mathcal{F}}$ which are the outputs of Algorithm 3 and Algorithm 2, respectively. Let $\widetilde{\mu}$ be the output of Lemma 21 (i.e., a size estimate of a random greedy maximal matching of H) with $\varepsilon k/2$ additive error. Then, it holds that

$$\frac{1}{2}\left(|\mathcal{U}_{low}| - \mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}})\right) - \frac{\varepsilon k}{2} \leq \widetilde{\mu} \leq |\mathcal{U}_{low}| - \mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}}).$$

Proof. Let M be any maximal matching of H. First, we show that

$$\frac{1}{2}\left(|\mathcal{U}_{low}| - \mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}})\right) \leq |M| \leq |\mathcal{U}_{low}| - \mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}}).$$

If for each edge in M, we take the corresponding set, and for the rest of the elements we take a set that only covers that element, we will have covered all elements of $\mathcal{U}_{low}$ with at most $|\mathcal{U}_{low}| - |M|$ sets, which implies that $|M| \leq |\mathcal{U}_{low}| - \mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}})$. On the other hand, no set can cover two elements that are unmatched by M at the same time. Thus, we have $|\mathcal{U}_{low}| - 2|M| \leq \mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}})$.

Since the output of *Lemma* 21 can be smaller than the minimum-sized maximal matching by an additive factor of at most $\varepsilon k/2$, we get the claim. ◁

▷ Claim 17. Conditioning on the high-probability events of Claim 11 and Claim 15, it holds that $\mathsf{SC}(\mathcal{U}, \mathcal{F}) \leq \mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}}) + \varepsilon k/2$.

Proof. All elements of $\mathcal{U} \setminus (\mathcal{U}_{low} \cup \mathcal{U}_{high})$ can be covered using the c sets that are deleted in Algorithm 2. Moreover, by Claim 11, we have $c < o(k)$ since $\alpha > \omega(1)$. On the other hand, the elements of $\mathcal{U}_{high}$ can be covered using $\varepsilon k/5$ sets, as shown in Claim 15. Finally, the elements of $\mathcal{U}_{low}$ can be covered using $\mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}})$ sets. Therefore, the union of these three collections covers all elements, and the size of this solution is at most $\mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}}) + \varepsilon k/2$, which completes the proof. ◁

▶ **Lemma 18.** *Conditioning on the high-probability events of Claim 11 and Claim 15, it holds that* $\chi/2 - \varepsilon k \leq \widetilde{\chi} \leq \chi$. *(Recall that* $\chi = k - \mathsf{SC}(\mathcal{U}, \mathcal{F})$ *and* $\widetilde{\chi}$ *is the output of Algorithm 1, defined on its Algorithm 1.)*

Proof. We have that

$$\begin{aligned}
\widetilde{\chi} &= \widetilde{\mu} + |\mathcal{U} \setminus \mathcal{U}_{low}| - \frac{\varepsilon k}{2} && \text{(Output of Algorithm 1)}\\
&\leq |\mathcal{U}_{low}| - \mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}}) + |\mathcal{U} \setminus \mathcal{U}_{low}| - \frac{\varepsilon k}{2} && \text{(By Claim 16)}\\
&= |\mathcal{U}| - \mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}}) - \frac{\varepsilon k}{2}\\
&\leq |\mathcal{U}| - \mathsf{SC}(\mathcal{U}, \mathcal{F}) && \text{(By Claim 17)}\\
&= \chi.
\end{aligned}$$

On the other hand,

$$\begin{aligned}
2(\widetilde{\chi} + \varepsilon k) &= 2\left(\widetilde{\mu} + |\mathcal{U} \setminus \mathcal{U}_{low}| + \frac{\varepsilon k}{2}\right)\\
&\geq 2\left(\frac{|\mathcal{U}_{low}|}{2} - \frac{\mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}})}{2} + |\mathcal{U} \setminus \mathcal{U}_{low}|\right) && \text{(By Claim 16)}\\
&\geq |\mathcal{U}| - \mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}})\\
&= |\mathcal{U}| - \mathsf{SC}(\mathcal{U}_{low}, \mathcal{F})\\
&\geq |\mathcal{U}| - \mathsf{SC}(\mathcal{U}, \mathcal{F})\\
&= \chi,
\end{aligned}$$

which concludes the proof. ◀

4.2 Time Complexity

▷ Claim 19. Algorithm 2 runs in $O(nk/\alpha)$ time with high probability.

Proof. First note that Algorithm 2 samples at most k/α elements for each set in $\mathcal{F}$ (Algorithm 2). So if we ignore the block of if-condition in Algorithm 2, the runtime is at most $O(nk/\alpha)$. Further, by Claim 11, the condition on Algorithm 2 holds at most k/α times with high probability. Since each time the algorithm enters the if-condition it spends $O(k)$ time to iterate over all elements, the total running time is at most $O(k^2/\alpha + nk/\alpha) = O(nk/\alpha)$. ◁

▷ Claim 20. Algorithm 3 runs in $O(k^2/\beta)$ time.

Proof. The algorithm samples $r_2 = O(k/\beta)$ sets and for each of them makes a membership query between the set and all elements. Hence, the total running time is upper-bounded by $O(k^2/\beta)$. ◁

We defer the proof of the next lemma to the full version of the paper.

▶ **Lemma 21.** *Let H be the multigraph defined in Definition 9. There exists an algorithm with an expected running time of $\widetilde{O}_\varepsilon(k\beta + \alpha\beta n)$ that estimates the value of $E_\pi|\mathrm{RGMM}(H,\pi)|$ with εk additive error with high probability.*

▶ **Lemma 22.** *The total running time of the algorithm is $\widetilde{O}(nk/\alpha + k^2/\beta + n\alpha\beta)$ with high probability.*

Proof. The algorithm runs in expected time $\widetilde{O}(nk/\alpha + n\alpha\beta)$ as shown by Claim 19, Claim 20, Lemma 21, and the fact that $k \le n$. To ensure a high-probability bound on the running time, we execute $\Theta(\log n)$ instances of the algorithm in parallel and use the estimate from the first instance that finishes. Since the expected running time is $\widetilde{O}(kn/\alpha + k^2/\beta + n\alpha\beta)$, the first instance is likely to terminate within $\widetilde{O}(nk/\alpha + k^2/\beta + n\alpha\beta)$ time with probability $1 - 1/\operatorname{poly}(n)$. (We remark that since our approximation guarantees hold with high probability, they also hold with high probability for each of the instances.) ◀

4.3 Putting Everything Together

We combine our ideas to obtain our final theorem for estimating $\chi = k - \mathsf{SC}(\mathcal{U},\mathcal{F})$. Then, we extend our analysis and make slight modifications to the algorithm to estimate $k - \mathsf{SC}(\mathcal{U},\mathcal{F}_{\neq 2})$ which is crucial for designing a sublinear algorithm for the Steiner tree problem.

▶ **Theorem 3** (Our Algorithm for Threshold Set Cover)**.** *There exists an algorithm that, given a set system $(\mathcal{U},\mathcal{F})$ with oracle access to its adjacency matrix (also known as membership queries), outputs a multiplicative-additive $(1/2, \varepsilon\cdot|\mathcal{U}|)$-approximation to Threshold Set Cover, in $\widetilde{O}(|\mathcal{F}|^{5/3})$ time, with high probability.*

Proof. First, if $k \le O(n^{2/3})$, then we can easily query between all elements and sets and compute the estimate in $\widetilde{O}(n^{5/3})$ time since all the steps of the algorithm such as finding the maximal matching can be run in linear time with respect to the size of the input. Now suppose that $k > n^{2/3}$. We use Algorithm 1 to produce the estimate $\widetilde{\chi}$, setting $x = 1/3$ and $y = 1/3$. By Lemma 18, we have that $\widetilde{\chi}$ is a multiplicative-additive $(1/2, \varepsilon k)$-approximation to the value of χ.

Moreover, we have $\alpha = n^{1/3}$ and $\beta = 10\max(k/n^{1-y}, 1)\cdot n\log(n)/k = \widetilde{O}(n^{1/3})$. By Lemma 22, the total runtime of the algorithm is upper-bounded by $\widetilde{O}(kn/\alpha + k^2/\beta + n\alpha\beta) = \widetilde{O}(n^{5/3})$ because of the assumption that $k \le n$. ◀

Now we show how we can estimate $k - \mathsf{SC}(\mathcal{U}, \mathcal{F}_{\neq 2})$ with a slight modification. The only change that is needed in our algorithm is to remove edges of H that are produced by sets of size 2 in $\hat{\mathcal{F}}$. Hence, when the algorithm estimates $\mathbf{E}_\pi[\mathrm{RGMM}(H, \pi)]$ using the vertex oracle, the oracles must not use those edges in the exploration as they do not exist in graph H. Thus, in the implementation of Lemma 21, when the algorithm queries pairs (u, S) where $u \in \mathcal{U}_{low}$ and $S \in \mathcal{F}_v$ to find neighbors of v in H, if $u \in S$, the algorithm starts querying between all elements of $\mathcal{U} \setminus v$ and S until it finds another element in S. If the algorithm finds another element of S, then it accepts the edge (v, u) as a valid edge, and otherwise it continues the random search. So each time the algorithm finds such pair (u, S), it invokes the above procedure to validate the edge. In the next theorem, we demonstrate how to bound the running time of the modified algorithm and discuss its approximation ratio.

▶ **Theorem 23.** *There exists an algorithm that outputs a multiplicative-additive $(1/2, \varepsilon k)$-approximation of value of $\chi = k - \mathsf{SC}(\mathcal{U}, \mathcal{F}_{\neq 2})$ in $\widetilde{O}(|\mathcal{F}|^{5/3})$ time with high probability.*

Proof. We need to show that the new estimation is a multiplicative-additive $(1/2, \varepsilon k)$-approximation of $k - \mathsf{SC}(\mathcal{U}, \mathcal{F}_{\neq 2})$. We follow the same approach as proof of Lemma 18. To follow the same steps, we need analogous claims similar to Claim 16 and Claim 17. The exact same proof of Claim 16 also works to provide a bound on $\mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}}_{\neq 2})$. More specifically, we have

$$\frac{1}{2}\left(|\mathcal{U}_{low}| - \mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}}_{\neq 2})\right) - \frac{\varepsilon k}{2} \leq \widetilde{\mu} \leq |\mathcal{U}_{low}| - \mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}}_{\neq 2}).$$

Further, to show that $\mathsf{SC}(\mathcal{U}, \mathcal{F}_{\neq 2}) \leq \mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}}_{\neq 2}) + \varepsilon k/2$, note that we have $c = o(k)$ (see Algorithm 2 for definition of c) which implies that $\mathcal{U} \setminus (\mathcal{U}_{low} \cup \mathcal{U}_{high})$ can be covered using $o(k)$ sets of size larger than 2 (all sets that are removed in Algorithm 2 have size larger than 2). Additionally, elements of $\mathcal{U}_{high}$ can be covered using $\varepsilon k/5$ sets of $\hat{\mathcal{F}}$ according to Claim 15. Thus, we can cover elements of $\mathcal{U}_{high}$ using $2\varepsilon k/5$ sets of $\hat{\mathcal{F}}_{\neq 2}$ since we can replace each set of size 2 with two sets of size 1 that cover a single element. Finally, elements of $\mathcal{U}_{low}$ can be covered using $\mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}}_{\neq 2})$ sets. The union of all these sets is a valid solution for $\mathsf{SC}(\mathcal{U}, \hat{\mathcal{F}}_{\neq 2})$ which has a size of at most $\mathsf{SC}(\mathcal{U}_{low}, \hat{\mathcal{F}}_{\neq 2}) + \varepsilon k/2$. Therefore, the error of our estimation is $(1/2, \varepsilon k)$.

Now it remains to bound the running time of the modified algorithm. We prove that the same bound on the running time as in Lemma 21 holds. Now consider a vertex v that is corresponding to an element in $\mathcal{U}_{low}$. Let $\hat{\mathcal{F}}_v$ be the collection of sets that include v. Also, let $\hat{\mathcal{F}}'_v = \{S | S \in \hat{\mathcal{F}}_v \text{ and } |S| = 2\}$. Let $r = |\hat{\mathcal{F}}_v|$ and $\hat{\mathcal{F}}_v = \{S_1, \ldots, S_r\}$. Also, let $\tau_i = |S_i \cap (\mathcal{U}_{low} \setminus v)|$. Hence, in Lemma 21, when the algorithm queries a pair (u, S) and it returns $u \in S$, the probability that $S = S_i$ is $\tau_i / \sum_{i=1}^{r} \tau_i$. For this set, using a Chernoff bound, the algorithm needs to spend at most $\widetilde{O}(k/\tau_i)$ time to see another element of S_i or explore all elements of $\mathcal{U}$. Therefore, the expected additional time the algorithm needs to spend compared to Lemma 21 is

$$\sum_{i}^{r} \left(\frac{\tau_i}{\sum_{i=1}^{r} \tau_i}\right) \cdot \widetilde{O}\left(\frac{k}{\tau_i}\right) = \widetilde{O}\left(\sum_{i}^{r} \frac{k}{\sum_{i=1}^{r} \tau_i}\right) \leq \widetilde{O}\left(\frac{rk}{\deg_H(v)}\right),$$

where the last inequality follows by the fact that $\deg_H(v) \leq \sum_{i=1}^{r} \tau_i$ since $\deg_H(v) = (\sum_{i=1}^{r} \tau_i) - |\{S | S \in \hat{\mathcal{F}}_v \text{ and } |S| = 2\}|$ after the modification of the graph H. On the other hand, the expected number of times that the oracle calls an adjacent edge of vertex v, is at most $\widetilde{O}(\deg_H(v)/k)$ (for the formal statement and the proof, see the full version of the paper). Also, these two variables have a negative correlation. Therefore, the total cost for all

vertices is at most $\widetilde{O}(rk)$. Combining with the fact that $r = \widetilde{O}(\beta)$ (Lemma 14), the total additional cost is $\widetilde{O}(k\beta)$, which is dominated by other terms in Lemma 22 which implies that the algorithm has the same running time as Theorem 3 if we choose α and β similarly. ◀

5 Connection to Steiner Tree

In this section, we show how our improved algorithm for set cover (from Section 4) implies an improved sublinear algorithm for metric Steiner tree. Formally, we show the following theorem.

▶ **Theorem 4** (Sublinear Algorithm for Metric Steiner Tree). *There exists an algorithm that, given an instance of metric Steiner tree denoted by (V, T, w) with oracle access $\mathcal{O}$ to the distance matrix of (V, w), outputs a $(2-\eta)$-estimate of $\mathsf{ST}(V, T, w)$ using $\widetilde{O}(n^{5/3})$ queries to $\mathcal{O}$, where $\eta > 0$ is a universal constant, with high probability.*

The overall structure of our algorithm is similar to the algorithm of Chen, Khanna and Tan [16]. The main difference in our algorithm compared to [16] is in the set cover component. In the following, we first provide an overview of their algorithm in Section 5.1. We then provide the query-efficient implementation of their algorithm and our modification to it in Section 5.2. We will finally provide the query complexity analysis and the proof of Theorem 4 in Section 5.3. Note that the approximation analysis of our algorithm follows directly from the proof of Theorem 3 in [16].

5.1 Algorithm at a High Level

Step 1: Minimum spanning tree over terminals. The algorithm of [16] as a first step starts with an MST $\mathcal{T}^*$ over the terminals T (whose cost can be estimated in nearly linear time using the sublinear MST algorithm of Czumaj and Sohler [18]). It is known that $w(\mathcal{T}^*)/2 \leq \mathsf{ST}(V, T, w) \leq w(\mathcal{T}^*)$. To get a strictly better-than-2 approximation of $\mathsf{ST}(V, T, w)$, it suffices to detect whether $\mathsf{ST}(V, T, w)$ is closer to $w(\mathcal{T}^*)/2$ or $w(\mathcal{T}^*)$. Hence, the rest of the algorithm is either to provide "significant" local improvements over $w(\mathcal{T}^*)$ using "set cover" like structure (i.e., step 2 in Section 4 of [16]) or "local structure" (i.e., step 3 in Section 4 of [16]), and thus output $(1 - O(\eta))w(\mathcal{T}^*)$ as the estimate of $\mathsf{ST}(V, T, w)$; or conclude that $\mathsf{ST}(V, T, w)$ is closer to $w(\mathcal{T}^*)$ and output it as the estimate of $\mathsf{ST}(V, T, w)$. Then, they show how to implement these steps using sublinear queries to $\mathcal{O}$.

Step 2: Improvement using set cover. First, they partition the edges into $L = O((\log k)/\varepsilon)$ buckets such that the edges of the ith bucket have weights in $[(1+\varepsilon)^{i-1}, (1+\varepsilon)^i)$. Let H_i be the graph built on all the terminals and all the edges upto the ith bucket. They define an instance of set cover corresponding to each level i where *ideally*,

- The elements correspond to the connected components of H_{i-1}.
- The sets correspond to the Steiner vertices.
- A set W_v (corresponding to a Steiner vertex v) contains an element u_S (corresponding to a component S) if the distance between v and some terminal in S is less than a threshold τ (think of it as $\frac{3}{5} \cdot (1+\varepsilon)^i$).

How to use set cover in making a decision about the Steiner tree cost. They show that if one can solve each of these set cover instances approximately, then one can check whether the total contribution of these set cover improvements is more than $O(\eta) \cdot w(\mathcal{T}^*)$. In particular, in that case $\mathsf{ST}(V, T, w)$ is strictly less than $(1 - O(\eta)) \cdot w(\mathcal{T}^*)$. Intuitively, this

is because one can include the Steiner vertices corresponding to the set cover solution and remove a subset of the edges in $\mathcal{T}^*$, while still maintaining a feasible solution to the Steiner tree instance. Hence, this implies that the cost of the constructed solution of the Steiner tree instance is less than $(1 - O(\eta)) \cdot w(\mathcal{T}^*)$.

A challenge and the notion of representatives. The main challenge with the above algorithm is computing the set cover instance. More precisely, in the third bullet point above, in order to check whether a set (corresponding to a Steiner vertex v) contains an element (corresponding to a connected component S), they need to compute the distance of v to all terminals $t \in T$ which could be very costly. Instead, they define a *net* on S which is a maximal subset $\tilde{S} \subseteq S$ such that any pair of terminals in $\tilde{S}$ has distance at least $\varepsilon \cdot (1+\varepsilon)^i$. They call the terminals in $\tilde{S}$ the *representatives*.

Modified set cover instance and the notion of light/heavy levels. Now, to detect if a set contains an element, we only need to check the distance of v to all terminals in $\tilde{S}$. When $|\tilde{S}|$ is small, this can be done efficiently. So, in their algorithm they *only* assign an element to a connected component S if the size of its representatives is *small*. To show that this does not introduce a large error, they define a level i to be *light* if the total sum of the edges of $\mathcal{T}^*$ in bucket i is "small", and define it to be *heavy* otherwise. They show that one can ignore all the levels i that are light, and moreover if a level is heavy, then most of its components have small sets of representatives and thus the error introduced by ignoring the components S with large net size is negligible.

Finally, we note that the notion of representatives will be used in other parts of the overall algorithm such as computing $\mathcal{T}^*$ which we will go over when describing the implementation of this step.

Step 3: Improvement using "local structure". In this step, they consider the hierarchical structure of the connected components. Specifically, they focus on components S that have exactly two child components, S_1 and S_2, where each of these child components also has exactly two child components: S_{11}, S_{12} for S_1 and S_{21}, S_{22} for S_2. Then, they check whether there exists a single Steiner vertex v that can be used to connect components $S_{11}, S_{12}, S_{21}, S_{22}$ and instead remove the corresponding edges in $\mathcal{T}^*$ connecting these components. Similarly to step 2, if the overall advantage of all these 2-level local improvements is more than $O(\eta) \cdot w(\mathcal{T}^*)$, then the algorithm outputs $(1 - O(\eta)) \cdot w(\mathcal{T}^*)$ as its estimate of $\mathsf{ST}(V, T, w)$.

Given that our algorithm does not change this step at all, we refer the reader to [16] for further details. Moreover, the query complexity is exactly the same as in [16].

5.2 Implementation of the Algorithm

Here, we focus on a query-efficient implementation of the algorithm, and particularly highlight where the set cover component was used and how we modify it.

First, we note that one cannot compute H_i exactly, so [16] shows that it suffices to work with an approximate graph H'_i such that $H_i \subseteq H'_i \subseteq H_{i+1}$.

Subroutines. Next, [16] define some useful subroutines for simulating the set cover instance on H'_i:

- $\mathsf{Find}(u, i)$: This receives a terminal u and a level i, and finds some representative terminal u' in the same connected component of H'_i that contains u. Moreover, they show that this subroutine can be implemented using $\widetilde{O}(k)$ queries.

- $\mathsf{BFS}(u, i)$: This subroutine reports all representative terminals that are in the same connected component in H'_i as u, if the total number of such representatives is $\widetilde{O}(L/\varepsilon) = \widetilde{O}(1/\varepsilon)$. Otherwise, the procedure is terminated. This procedure employs Find subroutines and has total query complexity of $\widetilde{O}(k/\varepsilon)$ which is $\widetilde{O}(k)$ given that ε is a constant.

Parameters. The algorithm uses four parameters, whose values will be later set to optimize the query complexity of the algorithm.

- M is a threshold parameter used on the number of components that contain a "small" number of representatives. Note that these are the components to which we assign an element in the universe $\mathcal{U}_i$ of the corresponding set cover instance $(\mathcal{U}_i, \mathcal{F}_i)$.
- R is a threshold parameter defining "low-degree" and "high-degree" elements.
- P is a threshold parameter denoting "low-degree" or "high-degree" sets.
- κ is a threshold parameter on the value of k. At a high level, when $k < \kappa$, we can afford to query all distances between terminals and the Steiner vertices.

Simulation of Step 2 and its query complexity. We now outline the implementation of Step 2 and specify the query complexity of each step, along with potential conditions they impose on the parameters we need to set.

- **The case of small number of terminals: if** $k \leq \kappa$, then we query all distances between terminals and Steiner vertices, which takes $O(n\kappa)$ queries. Then, we estimate $|\mathcal{U}| - \mathsf{SC}(\mathcal{U}, \mathcal{F})$ using our algorithm, but without any further queries.
- **Otherwise,** for each level i, they show that one of the following cases hold:

  Case 1. The total number of representative terminals in all connected components is $\widetilde{O}(M/\varepsilon)$. To detect this case, they use greedy MIS which can be implemented by the BFS and Find subroutines and will take $\widetilde{O}(Mk/\varepsilon)$ queries (for further details, see [43]). If this is the case, then again the set cover instance can easily be computed by querying the distance of all Steiner vertices to all representative terminals which requires $\widetilde{O}(nM/\varepsilon)$ queries. Then, similarly to the case of $k \leq \kappa$, $|\mathcal{U}_i| - \mathsf{SC}(\mathcal{U}_i, \mathcal{F}_i)$ can be estimated using our algorithm, without any further queries.

  Case 2. $|\mathcal{U}_i| \leq M$. In this case they show that level i is in fact light and thus can be ignored. To detect this case, they estimate the size of $|\mathcal{U}_i|$ using calls to BFS starting from $\widetilde{O}(k/M)$ random terminals, which overall takes $\tilde{O}(k^2/M)$ queries. Note that if we are in this case, we take no further action. Also, this step requires $k > M$, which we will ensure in our parameter setup.

  Case 3. The last case is when $|\mathcal{U}_i| \geq M$.
  - **Partitioning of the terminals based on their degree.** First, they partition the terminals into T_{low} and T_{high} based on whether the number of "close-by" Steiner vertices (roughly within distance $(3/5)(1+\varepsilon)^i$) to them is smaller than or larger than R. This partitioning can be computed using $\widetilde{O}(kn/R)$ queries by randomly sampling $\widetilde{O}(n/R)$ Steiner vertices and checking their distance to all the terminals.
  - **Handling high-degree terminals.** Then, by picking $\widetilde{O}(n/R)$ sets uniformly at random, with high probability, all elements corresponding to the components containing at least one terminal in T_{high} are covered. As they can only afford an $\varepsilon|\mathcal{U}_i|$ additive error in their estimate of $|\mathcal{U}_i| - \mathsf{SC}(\mathcal{U}_i, \mathcal{F}_i)$, they require that $n/R < \widetilde{O}(\varepsilon M) = \widetilde{O}(\varepsilon|\mathcal{U}_i|)$.

- **Handling low-degree terminals.** Next, they solve the set cover instance on $\mathcal{U}_{low}$, i.e., the connected components that have no terminal in T_{high}.
  [label=•, leftmargin=*]
  - **Partitioning of the Steiner vertices based on their degree.** They partition the sets of $\mathcal{F}_i$ into $\mathcal{W}_1$ and $\mathcal{W}_2$ based on whether their degree to T_{low} is less than $\Theta(P)$ or higher. This partitioning can be computed using $\widetilde{O}(nk/P)$ queries by randomly sampling k/P terminals from T_{low}. This requires $k > \widetilde{\Omega}(P)$ which we will ensure in our parameter setup. Then, they consider the set cover instances $(\mathcal{U}_{low}, \mathcal{W}_1)$ and $(\mathcal{U}_{low}, \mathcal{W}_2)$ separately, and return the better of the two solutions.

    Our modification to this step: In our set cover algorithm, however, we define $\mathcal{W}_2$ slightly differently, as described in Set Sparsification, see Algorithm 2. More precisely, we iterate over Steiner vertices (i.e., sets in $\mathcal{F}_i$) one by one in an arbitrary order, and at every round $j \leq |\mathcal{F}_i|$, we check whether the degree of the Steiner vertex v_j to T_{low} is more than $\Theta(P)$. If so, we add its corresponding set, W_j, to $\mathcal{W}_2$. Similarly to their test, our test can also be implemented using $\widetilde{O}(nk/P)$ queries. However, each time we add a set W_j to $\mathcal{W}_2$, we find all its "nearby" terminals and remove them from T_{low}, more precisely, $T_{low} \leftarrow T_{low} \setminus W_j$. This step can be simply done by querying the distance of the Steiner vertex v_j and all terminals in T_{low}. Hence, each time a set is added to $\mathcal{W}_2$, we perform an extra $\widetilde{O}(k)$ queries compared to the algorithm of [16]. However, as we can simply bound $|\mathcal{W}_2|$ by k/P, the overall query complexity remains as $\widetilde{O}(nk/P + k^2/P) = \widetilde{O}(nk/P)$.

  - **Handling high-degree Steiner vertices.** To solve $(\mathcal{U}_{low}, \mathcal{W}_2)$, note that by a simple double-counting argument, $|\mathcal{W}_2| \leq kR/P$. So if $kR/P \leq \varepsilon M \leq \varepsilon|\mathcal{U}_i|$, which will be ensured in the parameter setting, we can afford to pick all sets in $\mathcal{W}_2$ and thus, similarly to their argument, we only need to estimate $|\bigcup_{W\in\mathcal{W}_2} W|$ in the set cover instance. This is done by randomly sampling the terminals and using BFS and will take an overall $\widetilde{O}(k^2/M)$ queries.

    Our modification to this step: With the adjusted partitioning of $\mathcal{F}_i$ into $\mathcal{W}_1$ and $\mathcal{W}_2$ in our algorithm, the size of $\mathcal{W}_2$ is at most k/P. Therefore, by setting $k/P \leq \varepsilon M \leq \varepsilon|\mathcal{U}_i|$, we can afford to select all sets in $\mathcal{W}_2$. Notably, this modification relaxes the required condition of [16] from "$kR/P \leq \varepsilon M$" to "$k/P \leq \varepsilon M$".

  - **Handling low-degree Steiner vertices.** Finally, we need to solve $(\mathcal{U}_{low}, \mathcal{W}_1)$. In their approach, this part takes $O(RP \cdot RPk)$ queries, and this is the step where our main improvement comes from.

    Our modification to this step. By our improved bound for set cover (from Section 4), the query complexity of this part reduces to $\widetilde{O}(k^2/M + RP(n+k))$. More precisely, to simulate the algorithm in Lemma 21, we do the following.
    1. To run the RGMM oracle, we need to sample $\widetilde{O}(1)$ elements from $\mathcal{U}_i$ uniformly at random. Note that each element of $\mathcal{U}_i$ corresponds to a small component (a component with a small number of representatives). To find a small component uniformly at random, we first pick a terminal uniformly at random and run a BFS to determine whether it lies in a small component, and if so, whether it is a representative terminal. If the terminal is a representative and lies in a small component, we choose the corresponding connected component with probability $1/z$ where z is the number of representative terminals in that connected component (note that BFS returns this number as well). This approach ensures that each

small connected component has an equal probability of being sampled. Moreover, due to the bound on the number of small connected components, i.e., $|\mathcal{U}_i| \geq M$, we expect to encounter a terminal in a small connected component every $\widetilde{O}(k/M)$ samples. Therefore, the total cost of running all these BFS subroutines is $\widetilde{O}(k^2/M)$, since each BFS takes $\widetilde{O}(k)$ time.

2. For a small component (a vertex in graph H of Lemma 21), we need to identify all the Steiner nodes within a distance of at most τ (which is set roughly as $(3/5)(1+\varepsilon)^i$). This step can be completed in $\widetilde{O}(n)$ time. A similar step with the same time complexity also appears in the proof of Lemma 21.
3. Note that the number of Steiner nodes within this close distance is at most R. Let $\hat{\mathcal{S}}$ be the set of these Steiner nodes. Next, the RGMM algorithm requires a random neighbor of a small component. To get such a neighbor, we keep picking pairs (v,t) in a random order, where $t \in T_{low}$ and $v \in \hat{\mathcal{S}}$, and querying their distance. The first time that we find a pair (v,t) in close distance, we run a BFS from t to check if it is a representative terminal and if it lies in a small component, which takes $\widetilde{O}(k)$ time. If it is not a representative terminal or does not lie in a small connected component, we skip this terminal. Otherwise, we return its small connected component with probability $1/z$, where z is the number of representative terminals in that connected component, ensuring that all neighbors have an equal probability of being selected. We run the above procedure until we find a random neighbor. Therefore, using the running time from Lemma 21 (substituting α and β for R and P), and considering that we run the BFS at most $\widetilde{O}(RP)$ times (which corresponds to the maximum degree of H), we obtain an $\widetilde{O}(RP(n+k))$ time algorithm to estimate the size of the matching.

Simulation of Step 3 and its query complexity. As we are following the exact implementation of [16] for this step, the additional query complexity of this step (compared to the Step 2) is equal to $\widetilde{O}(nk/M)$ for both their algorithm and our algorithm.

5.3 Query Complexity Analysis and Proof of Theorem 4

For completeness, we start with the query complexity analysis of [16].

Analysis of the query complexity of the algorithm of [16]. As computed in Section 5.2, the overall query complexity of their algorithm is bounded by

$$\begin{aligned}&\widetilde{O}(n\kappa + \frac{Mk}{\varepsilon} + \frac{nM}{\varepsilon} + \frac{k^2}{M} + \frac{nk}{R} + \frac{nk}{P} + \frac{k^2}{M} + (RP)^2 k + \frac{nk}{M})\\ &= \widetilde{O}(n\kappa + nM + \frac{nk}{R} + \frac{nk}{P} + (RP)^2 k + \frac{nk}{M}). && \triangleright\ \varepsilon = O(1), k \leq n\end{aligned}$$

Furthermore, the conditions that need to be satisfied are

- $k \leq \kappa$, or
- $k > M$ and $n/R < \widetilde{O}(\varepsilon M)$ and $k > \widetilde{\Omega}(P)$ and $kR/P \leq \varepsilon M$.

The query complexity of their algorithm under the above conditions can be optimized by setting $\kappa = M = n^{6/7}$, $R = n^{1/7}$, and $P = n^{2/7}$, which gives the total query complexity of $\widetilde{O}(n^{13/7})$.

Now, we prove the main theorem of this section.

Proof of Theorem 4. As we are implementing the same algorithm as [16], except replacing their set cover subroutine with a more efficient algorithm, the approximation analysis follows exactly from their proof. It only remains to bound the query complexity of our proposed algorithm for metric Steiner tree using the improved sublinear algorithm for set cover. In Section 5.2, we analyzed the query complexity of the component that is implemented differently in our algorithm. Now, we put the query complexity of all parts together and compute the overall complexity.

Analysis of the query complexity of our algorithm. The overall query complexity of our algorithm is bounded by

$$\begin{aligned}&\widetilde{O}(n\kappa + \frac{Mk}{\varepsilon} + \frac{nM}{\varepsilon} + \frac{k^2}{M} + \frac{nk}{R} + \frac{nk}{P} + \frac{k^2}{M} + RP(k+n) + \frac{nk}{M})\\ &= \widetilde{O}(n\kappa + nM + \frac{nk}{R} + \frac{nk}{P} + RPn + \frac{nk}{M}) \qquad \triangleright\, \varepsilon = O(1), k \le n\end{aligned}$$

Note again that the main difference is that the term P^2R^2k is replaced by RPn. Furthermore, the conditions that need to be satisfied are also slightly more relaxed, as follows:

- $k \le \kappa$, or
- $k > M$ and $n/R < \widetilde{O}(\varepsilon M)$ and $k > \widetilde{\Omega}(P)$ and $k/P \le \varepsilon M$.

Specifically, "$kR/P \le \varepsilon M$" is replaced by "$k/P \le \varepsilon M$". Then, our algorithm can be optimized by setting $\kappa = M = n^{2/3}, R = \widetilde{\Theta}(P) = \widetilde{\Theta}(n^{1/3})$ which gives the total query complexity of $\widetilde{O}(n^{5/3})$. ◀

References

1. Sepehr Assadi. Tight space-approximation tradeoff for the multi-pass streaming set cover problem. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on principles of database systems*, pages 321–335, 2017. doi:10.1145/3034786.3056116.
2. Sepehr Assadi, Sanjeev Khanna, and Yang Li. Tight bounds for single-pass streaming complexity of the set cover problem. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 698–711, 2016. doi:10.1145/2897518.2897576.
3. Baruch Awerbuch, Yossi Azar, and Yair Bartal. On-line generalized steiner problem. *Theoretical Computer Science*, 324(2-3):313–324, 2004. doi:10.1016/J.TCS.2004.05.021.
4. Amir Azarmehr, Soheil Behnezhad, and Mohammad Roghani. Fully dynamic matching: -approximation in polylog update time. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 3040–3061. SIAM, 2024. doi:10.1137/1.9781611977912.109.
5. MohammadHossein Bateni, Hossein Esfandiari, and Vahab Mirrokni. Almost optimal streaming algorithms for coverage problems. In *Proceedings of the 29th ACM Symposium on Parallelism in Algorithms and Architectures*, pages 13–23, 2017. doi:10.1145/3087556.3087585.
6. Soheil Behnezhad. Time-optimal sublinear algorithms for matching and vertex cover. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 873–884, 2021. doi:10.1109/FOCS52979.2021.00089.
7. Soheil Behnezhad, Mohammad Roghani, and Aviad Rubinstein. Local computation algorithms for maximum matching: New lower bounds. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 2322–2335. IEEE, 2023. doi:10.1109/FOCS57990.2023.00143.
8. Soheil Behnezhad, Mohammad Roghani, and Aviad Rubinstein. Sublinear time algorithms and complexity of approximate maximum matching. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 267–280. ACM, 2023. doi:10.1145/3564246.3585231.

9 Soheil Behnezhad, Mohammad Roghani, and Aviad Rubinstein. Approximating maximum matching requires almost quadratic time. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 444–454. ACM, 2024. doi:10.1145/3618260.3649785.

10 Soheil Behnezhad, Mohammad Roghani, Aviad Rubinstein, and Amin Saberi. Beating greedy matching in sublinear time. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023*, pages 3900–3945. SIAM, 2023. doi:10.1137/1.9781611977554.CH151.

11 Soheil Behnezhad, Mohammad Roghani, Aviad Rubinstein, and Amin Saberi. Sublinear algorithms for TSP via path covers. In *51st International Colloquium on Automata, Languages, and Programming, ICALP*, volume 297 of *LIPIcs*, pages 19:1–19:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.ICALP.2024.19.

12 Sayan Bhattacharya, Peter Kiss, and Thatchaphol Saranurak. Dynamic $(1+\varepsilon)$-approximate matching size in truly sublinear update time. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 1563–1588. IEEE, 2023. doi:10.1109/FOCS57990.2023.00095.

13 Sayan Bhattacharya, Peter Kiss, Thatchaphol Saranurak, and David Wajc. Dynamic matching with better-than-2 approximation in polylogarithmic update time. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 100–128. SIAM, 2023. doi:10.1137/1.9781611977554.CH5.

14 Jaroslaw Byrka, Fabrizio Grandoni, Thomas Rothvoß, and Laura Sanita. An improved lp-based approximation for steiner tree. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 583–592, 2010. doi:10.1145/1806689.1806769.

15 Bernard Chazelle, Ronitt Rubinfeld, and Luca Trevisan. Approximating the minimum spanning tree weight in sublinear time. *SIAM Journal on computing*, 34(6):1370–1379, 2005. doi:10.1137/S0097539702403244.

16 Yu Chen, Sanjeev Khanna, and Zihan Tan. Query complexity of the metric steiner tree problem. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4893–4935. SIAM, 2023. doi:10.1137/1.9781611977554.CH179.

17 Miroslav Chlebík and Janka Chlebíková. The steiner tree problem on graphs: Inapproximability results. *Theoretical Computer Science*, 406(3):207–214, 2008. doi:10.1016/J.TCS.2008.06.046.

18 Artur Czumaj and Christian Sohler. Estimating the weight of metric minimum spanning trees in sublinear time. *SIAM Journal on Computing*, 39(3):904–922, 2009. doi:10.1137/060672121.

19 Erik D Demaine, Piotr Indyk, Sepideh Mahabadi, and Ali Vakilian. On streaming and communication complexity of the set cover problem. In *Distributed Computing: 28th International Symposium, DISC 2014, Austin, TX, USA, October 12-15, 2014. Proceedings 28*, pages 484–498. Springer, 2014. doi:10.1007/978-3-662-45174-8_33.

20 Yuval Emek and Adi Rosén. Semi-streaming set cover. *ACM Transactions on Algorithms (TALG)*, 13(1):1–22, 2016. doi:10.1145/2957322.

21 Manuela Fischer and Andreas Noever. Tight analysis of parallel randomized greedy MIS. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 2152–2160. SIAM, 2018. doi:10.1137/1.9781611975031.140.

22 Naveen Garg, Anupam Gupta, Stefano Leonardi, and Piotr Sankowski. Stochastic analyses for online combinatorial optimization problems. In Shang-Hua Teng, editor, *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 942–951, 2008. URL: http://dl.acm.org/citation.cfm?id=1347082.1347185.

23 Edgar N Gilbert and Henry O Pollak. Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 16(1):1–29, 1968.

24 Christoph Grunau, Slobodan Mitrovic, Ronitt Rubinfeld, and Ali Vakilian. Improved local computation algorithm for set cover via sparsification. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2993–3011. SIAM, 2020. doi:10.1137/1.9781611975994.181.

25 Anupam Gupta, MohammadTaghi Hajiaghayi, and Amit Kumar. Stochastic steiner tree with non-uniform inflation. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 134–148. Springer, 2007. doi:10.1007/978-3-540-74208-1_10.

26 Anupam Gupta and Amit Kumar. Online steiner tree with deletions. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 455–467. SIAM, 2014. doi:10.1137/1.9781611973402.34.

27 Anupam Gupta and Martin Pál. Stochastic steiner trees without a root. In *Automata, Languages and Programming: 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005. Proceedings 32*, pages 1051–1063. Springer, 2005. doi:10.1007/11523468_85.

28 Sariel Har-Peled, Piotr Indyk, Sepideh Mahabadi, and Ali Vakilian. Towards tight bounds for the streaming set cover problem. In Tova Milo and Wang-Chiew Tan, editors, *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 371–383. ACM, 2016. doi:10.1145/2902251.2902287.

29 Makoto Imase and Bernard M Waxman. Dynamic steiner tree problem. *SIAM Journal on Discrete Mathematics*, 4(3):369–384, 1991. doi:10.1137/0404033.

30 Piotr Indyk, Sepideh Mahabadi, Ronitt Rubinfeld, Jonathan Ullman, Ali Vakilian, and Anak Yodpinyanee. Fractional set cover in the streaming model. In *20th International Workshop on Approximation Algorithms for Combinatorial Optimization Problem (APPROX 2017)*, 2017.

31 Piotr Indyk, Sepideh Mahabadi, Ronitt Rubinfeld, Ali Vakilian, and Anak Yodpinyanee. Set cover in sub-linear time. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2467–2486. SIAM, 2018. doi:10.1137/1.9781611975031.158.

32 Kumar Joag-Dev and Frank Proschan. Negative association of random variables with applications. *The Annals of Statistics*, 11(1):286–295, 1983.

33 Michael Kapralov, Slobodan Mitrovic, Ashkan Norouzi-Fard, and Jakab Tardos. Space efficient approximation to maximum matching size from uniform edge samples. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1753–1772. SIAM, 2020. doi:10.1137/1.9781611975994.107.

34 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of Symposium on the Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-1-4684-2001-2_9.

35 Alam Khursheed and K. M. Lai Saxena. Positive dependence in multivariate distributions. *Communications in Statistics - Theory and Methods*, 10(12):1183–1196, 1981.

36 Reut Levi, Ronitt Rubinfeld, and Anak Yodpinyanee. Brief announcement: Local computation algorithms for graphs of non-constant degrees. In Guy E. Blelloch and Kunal Agrawal, editors, *Proceedings of the 27th ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2015, Portland, OR, USA, June 13-15, 2015*, pages 59–61. ACM, 2015. doi:10.1145/2755573.2755615.

37 Nicole Megow, Martin Skutella, José Verschae, and Andreas Wiese. The power of recourse for online mst and tsp. *SIAM Journal on Computing*, 45(3):859–880, 2016. doi:10.1137/130917703.

38 Krzysztof Onak, Dana Ron, Michal Rosen, and Ronitt Rubinfeld. A Near-Optimal Sublinear-Time Algorithm for Approximating the Minimum Vertex Cover Size. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1123–1131, 2012. doi:10.1137/1.9781611973099.88.

39 Michal Parnas and Dana Ron. Approximating the minimum vertex cover in sublinear time and a connection to distributed algorithms. *Theor. Comput. Sci.*, 381(1-3):183–196, 2007. doi:10.1016/J.TCS.2007.04.040.

40 Gabriel Robins and Alexander Zelikovsky. Tighter bounds for graph steiner tree approximation. *SIAM Journal on Discrete Mathematics*, 19(1):122–134, 2005. doi:10.1137/S0895480101393155.

41 Barna Saha and Lise Getoor. On maximum coverage in the streaming model & application to multi-topic blog-watch. In *Proceedings of the 2009 siam international conference on data mining*, pages 697–708. SIAM, 2009. doi:10.1137/1.9781611972795.60.

42 David Wajc. Negative association: definition, properties, and applications. *Manuscript, available from https://goo. gl/j2ekqM*, 2017.

43 Yuichi Yoshida, Masaki Yamamoto, and Hiro Ito. An improved constant-time approximation algorithm for maximum matchings. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC*, pages 225–234. ACM, 2009. doi:10.1145/1536414.1536447.

Gadgetless Lifting Beats Round Elimination: Improved Lower Bounds for Pointer Chasing

Xinyu Mao
Thomas Lord Department of Computer Science, University of Southern California, Los Angeles, CA, USA

Guangxu Yang
Thomas Lord Department of Computer Science, University of Southern California, Los Angeles, CA, USA

Jiapeng Zhang
Thomas Lord Department of Computer Science, University of Southern California, Los Angeles, CA, USA

Abstract

We prove an $\Omega(n/k + k)$ communication lower bound on $(k-1)$-*round distributional complexity* of *the k-step pointer chasing problem* under *uniform input distribution*, improving the $\Omega(n/k - k\log n)$ lower bound due to Yehudayoff (Combinatorics Probability and Computing, 2020). Our lower bound almost matches the upper bound of $\widetilde{O}(n/k + k)$ communication by Nisan and Wigderson (STOC 91).

As part of our approach, we put forth *gadgetless lifting*, a new framework that lifts lower bounds for *a family of restricted protocols* into lower bounds for *general protocols*. A key step in gadgetless lifting is choosing the appropriate definition of restricted protocols. In this paper, our definition of restricted protocols is inspired by the structure-vs-pseudorandomness decomposition by Göös, Pitassi, and Watson (FOCS 17) and Yang and Zhang (STOC 24).

Previously, round-communication trade-offs were mainly obtained by round elimination and information complexity. Both methods have some barriers in some situations, and we believe gadgetless lifting could potentially address these barriers.

2012 ACM Subject Classification Theory of computation → Communication complexity

Keywords and phrases communication complexity, lifting theorems, pointer chasing

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.75

Related Version *Full Version*: https://arxiv.org/abs/2411.10996
Full Version: https://eccc.weizmann.ac.il/report/2024/070/

Funding *Xinyu Mao*: Supported by NSF CAREER award 2141536.
Guangxu Yang: Supported by NSF CAREER award 2141536.
Jiapeng Zhang: Supported by NSF CAREER award 2141536.

Acknowledgements We thank Sepehr Assadi, Yuval Filmus and anonymous reviewers for their helpful comments.

1 Introduction

Pointer chasing is a well-known problem [26] that demonstrates the power of interaction in communication and has broad applications in different areas. It was used for proving monotone constant-depth hierarchy theorem [23, 20], lower bounds on the time complexity of distributed computation [22], lower bounds on the space complexity of streaming algorithms [10, 15, 1], adaptivity hierarchy theorem for property testing [5], exponential separations in local differential privacy [16], memory bounds for continual learning [7] and limitations of the transformer architecture [24]. It is a two-party function defined below.


© Xinyu Mao, Guangxu Yang, and Jiapeng Zhang;
licensed under Creative Commons License CC-BY 4.0



▶ **Definition 1** (k-step pointer chasing function). *For $k \geq 1$, the k-step pointer chasing function $PC_k : [n]^n \times [n]^n \to \{0,1\}$ is defined as follows. Given input $f_A, f_B \in [n]^n$, for $r = 0, 1, \ldots, k$ we recursively define pointers via*

$$pt_r(f_A, f_B) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } r = 0; \\ f_A(pt_{r-1}(f_A, f_B)) & \text{if } r > 0 \text{ is odd}; \\ f_B(pt_{r-1}(f_A, f_B)) & \text{if } r > 0 \text{ is even}. \end{cases}$$

The output of PC_k is the parity of the last pointer, namely, $PC_k(f_A, f_B) \stackrel{\text{def}}{=} pt_k(f_A, f_B) \bmod 2$.

Upper bounds

If Alice and Bob could communicate for k rounds, a simple protocol is the following: Alice and Bob alternatively send $f_A(\mathsf{pt}_{r-1}(f_A, f_B))$ or $f_A(\mathsf{pt}_{r-1}(f_A, f_B))$. The total communication cost for this simple protocol is $O(k \cdot \log n)$. However, if Alice and Bob can only communicate $(k-1)$ rounds, the upper bound then becomes non-trivial. Nisan and Wigderson [23] proposed a randomized $(k-1)$-round protocol with $O((n/k + k) \log n)$ communication bits.

- In the beginning, Alice and Bob use public randomness to pick a set of coordinates $I \subseteq [n]$ of size $10n/k$, and then send $f_A(I)$ and $f_B(I)$ to the other party.
- On the other hand, Alice and Bob also simulate (r rounds) deterministic protocol but skip one round if one party finds that the pointer is located in I.
- If the skip round never happens, Alice and Bob simply abort at the last round. A simple calculation shows the probability of this event is low.

This randomized protocol is indeed very simple. Alice and Bob only share coordinate-wise information. In fact, this is a structured rectangle in our setting.

Lower Bounds

Consider $(k-1)$ round protocols where Alice speaks first. For deterministic protocols, Nisan and Wigderson [23] proved an $\Omega(n - k \log n)$ communication lower bound. In the same paper, they also proved an $\Omega(n/k^2 - k \log n)$ communication lower bound for protocols that achieve $2/3$ accuracy *under uniform input distribution*.

Since then, lower bounds for pointer chasing and its close variants have been substantially studied by a good amount of papers [9, 8, 25, 18, 19, 10, 15, 1]. Finally, Yehudayoff [29] proved an $\Omega(n/k - k \log n)$ lower bound for protocols achieving constant advantage under uniform input distribution.

Now the main gap between the upper bound [23] and the lower bound [29] is the extra $k \log n$ term. This gap becomes significant if $k \geq \sqrt{n}$. In this paper, we further improve the lower bound and close the gap.

1.1 Our results

We prove that any protocol that achieves constant advantage under uniform input distribution must communicate $\Omega(n/k + k)$ bits.

▶ **Theorem 2.** *Let Π be a $(k-1)$-round deterministic protocol for PC_k where Alice speaks first such that*

$$\Pr_{f_A, f_B \leftarrow [n]^n} [\Pi(f_A, f_B) = PC_k(f_A, f_B)] \geq 2/3.$$

Then the communication complexity of Π is $\Omega(n/k + k)$.

By Yao's minimax principle, it implies a lower bound for the $(k-1)$ round randomized communication complexity.

▶ **Corollary 3.** *Every $(k-1)$-round randomized protocol for PC_k with error at most $1/3$ (where Alice speaks first) has communication complexity $\Omega(n/k+k)$.*

We observe there is still a $(\log n)$ gap between our lower bound and the protocol by [23]. We conjecture that our lower bound is tight and there is a chance to remove the $\log n$ factor in the upper bound side. A simple deterministic protocol with $(k-1)$ rounds and $O(n)$ communication bits could be the following: Alice and Bob send the parity of $f_A(x)$ and $f_B(x)$ for all $x \in [n]$ in the beginning. Hence they can skip the last round as they already know the parity. This simple protocol shows that [23]'s protocol is not tight when $k = o(\log n)$. We believe similar ideas could be extended for large k.

Applications

Given the connections between PC_k and diverse applications [10, 22, 5, 16, 7, 24], our improved lower bounds automatically lead to several applications. We list two applications below.

▶ **Corollary 4** (Direct sum extension of pointer chasing)**.** *The $(k-1)$-round randomized communication complexity of PC_k with d pairs of functions is $\Omega(d \cdot n/k^2 + d)$*

This corollary improves the previous $\Omega(d \cdot n/k^3 - dk\log n - 2d)$ lower bound presented in [10], which has applications in BFS trees streaming lower bound.

▶ **Corollary 5** (Exponential separations in local differential privacy)**.** *Let A be a $(k-1)$-round sequentially interactive ε-locally private protocol solving PC_k with error probability $\gamma \le 1/3$. Then the sample complexity of A is $\Omega\left(\frac{1}{e^\varepsilon} \cdot (n/k+k)\right)$ and there is a k round protocol with sample complexity $\widetilde{O}\left(\frac{k\log n}{\varepsilon^2}\right)$.*

This corollary improves the previous $\Omega\left(\frac{n}{e^\varepsilon k^2}\right)$ lower bound for $k < \sqrt{n/\log n}$ given by [16].

1.2 Gadgetless Lifting: A New Framework to Prove Communication Lower Bounds

The following two-step approach for proving communication lower bounds often appears in previous works (e.g., [11, 27]):

1. Identify a family of structured protocols.
2. Simulate general protocols by structured protocols and prove communication lower bounds for structured protocols.

This approach culminates in query-to-communication lifting theorems [12, 13, 6, 21].

Query-to-communication lifting theorems

Let $f : Z^n \to \{0,1\}$ be a function, and let $g : X \times Y \to Z$ be a two-party gadget function. The goal is to prove communication lower bounds for the function $f \circ g^n : X^n \times Y^n \to \{0,1\}$. Indeed, all functions for which lower bounds are proven using the above approach can be written as $f \circ g^n$ for appropriate f and g. For such functions, a communication protocol can always simulate a decision tree that computes f – such protocols consist of a natural family of structured protocols. Communication complexity for such protocols is essentially the query complexity of f, for which lower bounds are often easy to prove. Hence, the primary job is to show how to simulate general protocols by structured ones.

Though query-to-communication lifting is a beautiful framework, it requires a gadget function g since f is a one-party function. As a consequence, this framework only applies to *lifted functions*, namely, functions that can be written as $f \circ g^n$. Many important problems, such as pointer chasing, do not fall into this category; hence, lifting theorems do not apply in those cases.

To address this limitation, we propose a new framework called *gadgetless lifting*. We take a step back to the original approach, reconsidering the choice of structured protocols. In some cases, although the function is not a lifted function, there are simple and natural protocols. The crux of gadgetless lifting is how to decide the structured protocols. In this paper, we capture it as those protocols that "all shared useful information are local information". For example, the protocol by [23] only share local information such as $f_A(x)$ or $f_B(x)$ for some $x \in [n]$. In lemma 11, we show that any protocol for PC_k can be simulated by such protocols. Our proof is inspired by the structure-vs-pseudorandomness decomposition by Göös, Pitassi, and Watson [13] and Yang and Zhang [28], which is a powerful tool that emerged in the study of query-to-communication lifting theorems. Therefore, we call our method "gadgetless lifting".

In the study of lifted functions, it has been shown that query-to-communication lifting theorems bypassed some fundamental barriers from previous methods. Similarly, gadgetless lifting can also bypass obstacles from existing methods. We discuss two of them below.

Avoiding the loss in round elimination method

Previously, the only method to prove round-communication trade-offs is the *round elimination method* [23]. In [23] and [29], the authors studied the pointer chasing problem via the round elimination method. Denote by $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_t$ the messages sent in the first t rounds, and let $\boldsymbol{Z}_i$ be the pointer in the i-th round, i.e., , $Z_i = \mathsf{pt}_i(X, Y)$ where X, Y are uniformly chosen from $[n]^n$. As is standard the round elimination method, [23, 29] analyzed the random variables

$$\boldsymbol{R}_t = (\boldsymbol{M}_1, \ldots, \boldsymbol{M}_t, \boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_{t-1}) \text{ for } t \le k.$$

They proved that $\mathbf{H}(\boldsymbol{R}_k) \ge \Omega(n/k)$. Together with the fact that $\mathbf{H}(Z_1, \ldots, Z_k) = k \log n$, it implies that $\mathbf{H}(\boldsymbol{M}) \ge \Omega(n/k - k \log n)$. The $(k \log n)$ loss (or something similar) appears in many previous works that adopt round elimination-based [23, 18, 19, 14, 10, 29]. In this paper, we avoid the $k \log n$ loss via the gadgetless lifting.

Breaking square-root loss barrier in information complexity

Another popular method in proving communication lower bounds is by way of information complexity. However, as mentioned by Yahudayoff [29], entropy-based analyses are likely to induce a square-root loss barrier. This barrier usually comes from applying Pinsker's inequality (or its variant) to bound statistical distance from a small entropy gap. As a consequence, many results such as [23] can only prove an $\Omega(n/k^2 - k \log n)$ lower bound.

As mentioned in [29], the square-root loss also appears in many works when using the entropy-based method to prove lower bounds. For example, it appears in the parallel repetition theorem and is related to the "strong parallel repetition" conjecture which is motivated by Khot's unique games conjecture [17]. This loss also appears in direct-sum theorems [2] and direct-product theorems [4] in communication complexity.

[29] overcomes this square-root loss barrier by using a non-standard measurement called triangular discrimination. By contrast, our approach overcomes the barrier more naturally without using entropy.

Potential applications

We noticed that our method can also be naturally extended to multiparty settings such as the numbers in hand model. Moreover, some important open problems, such as round-communication tradeoff of bipartite matching problem [3] and set pointer chasing problem [10, 15], are difficult to solve using the round elimination method due to its inherent limitations. Our method offers the potential to solve these challenging problems.

2 Preliminaries

Notations

We use capital letters X to denote a set and use bold symbols like $\boldsymbol{R}$ to denote random variables. Particularly, for a set X, we use $\boldsymbol{X}$ to denote the random variable uniformly distributed over the set X. We use $\leftarrow$ to denote sampling from a distribution or choosing an element from a set uniformly at random.

2.1 Density-Restoring Partition

Min-entropy and dense distribution

For a random variable $\boldsymbol{X}$, we use $\mathrm{supp}(\boldsymbol{X})$ to denote the support of $\boldsymbol{X}$.

▶ **Definition 6** (Min-entropy and deficiency). *The min-entropy of a random variable $\boldsymbol{X}$ is defined by*

$$\mathbf{H}_\infty(\boldsymbol{X}) := \min_{x \in \mathrm{supp}(\boldsymbol{X})} \log\left(\frac{1}{\mathbf{Pr}[\boldsymbol{X} = x]}\right).$$

Suppose that $\boldsymbol{X}$ is supported on $[n]^J$. We define the deficiency of $\boldsymbol{X}$ as

$$\mathbf{D}_\infty(\boldsymbol{X}) := |J| \log n - \mathbf{H}_\infty(\boldsymbol{X}).$$

For $I \subseteq J$, $x \in [n]^J$, let $x(I) \stackrel{\mathrm{def}}{=} (x(i))_{i \in I} \in [n]^I$ be the projection of x on coordinates in I.

▶ **Definition 7** (Dense distribution). *Let $\gamma \in (0,1)$. A random variable $\boldsymbol{X}$ supported on $[n]^J$ is said to be γ-dense if for all nonempty $I \subseteq J$, $\mathbf{H}_\infty(x(I)) \geq \gamma |I| \log n$.*

The following lemma is the crux of the structure-vs-pseudorandomness method by [13]. It essentially says that a flat random variable could be decomposed into a convex combination of flat random variables with disjoint support and dense properties.

▶ **Lemma 8** (Density-restoring partition). *Let $\gamma \in (0,1)$. Let X be a subset of $[n]^M$ and $J \subseteq [M]$. Suppose that there exists an $\beta \in [n]^{\bar{J}}$ such that $\forall x \in X, x(\bar{J}) = \beta$. Then, there exists a partition $X = X^1 \cup X^2 \cup \cdots \cup X^r$ and every X^i is associated with a set $I_i \subseteq J$ and a value $\alpha_i \in [n]^{I_i}$ that satisfy the following properties.*

1. $\forall x \in X^i, x(I_i) = \alpha_i$;
2. $\boldsymbol{X}^i(J \setminus I_i)$ *is γ-dense;*
3. $\mathbf{D}_\infty\left(\boldsymbol{X}^i(J \setminus I_i)\right) \leq \mathbf{D}_\infty\left(\boldsymbol{X}(J)\right) - (1-\gamma)\log n \cdot |I_i| + \delta_i$, *where* $\delta_i \stackrel{\mathrm{def}}{=} \log(|X|/|\cup_{j \geq i} X^j|)$.

The proof of this lemma, simple and elegant, is included in the appendix for completeness.

2.2 Communication Protocols

We recall basic definitions and facts about communication protocols.

Protocol Tree

Let X and Y be the input space of Alice and Bob respectively. A deterministic communication protocol Π is specified by a rooted binary tree. For every internal vertex v,

- it has 2 children, denoted by $\Pi(v,0)$ and $\Pi(v,1)$;
- v is owned by either Alice or Bob – we denote the owner by $\mathsf{owner}(v)$;
- every leaf node specifies an output.

Starting from the root, the owner of the current node cur partitions its input space into two parts X_0 and X_1, and sets the current node to $\Pi(\mathsf{cur}, b)$ if its input belongs to X_b.

▶ **Fact 9.** *The set of all inputs that leads to an internal vertex v is a rectangle, denoted by $\Pi_v = X_v \times Y_v \subseteq X \times Y$.*

The *communication complexity* of Π, denoted by $\mathrm{CC}(\Pi)$, is the depth of the tree. The *round complexity* of Π, is the minimum number k such that in every path from the root to some leaf, the owner switches at most $(k-1)$ times. Clearly, if a protocol has k round, then its communication complexity is at least k. We can safely make the following assumptions for any protocol Π:

- Π has k rounds on every input; and
- Π communicates $\mathrm{CC}(\Pi)$ bits on every input.

Indeed, for any protocol, we can add empty messages and rounds in the end, which boosts the communication complexity by a factor of 2.

3 Proof of Main Theorem

▶ **Theorem 10** (Main theorem, Theorem 2 restated)**.** *Let Π be a $(k-1)$-round deterministic protocol for PC_k where Alice speaks first such that*

$$\Pr_{f_A, f_B \leftarrow [n]^n} [\Pi(f_A, f_B) = PC_k(f_A, f_B)] \geq 2/3.$$

Then the communication complexity of Π is $\Omega(n/k + k)$.

We use a *decomposition and sampling process* DS, as shown in Algorithm 1, in our analysis. DS takes as input a protocol Π, and samples a rectangle R that is contained in Π_v for some leaf node v. Our proof proceeds in three steps:

1. First, Section 3.1 analyzes crucial invariants during the running of DS.
2. Next, Section 3.2 shows that the accuracy of Π is captured by a quantity called *average fixed size*, which is a natural quantity that arises in the running of DS.
3. Finally, Section 3.3 proves that the average fixed size can be bounded from above by $O(\mathrm{CC}(\Pi))$. Consequently, if Π enjoys high accuracy, we get a lower bound of $\mathrm{CC}(\Pi)$.

3.1 The Decomposition and Sampling Process

During the sampling process, we maintain a useful structure of R mainly by a partitioning-then-sampling mechanism: At the beginning, R is set to be the set of all inputs. Walking down the protocol tree, we decompose the rectangle into structured sub-rectangles; then we sample a decomposed rectangle with respect to its size. In the end, we arrive at a leaf node v and a subrectangle of Π_v.

Algorithm 1 The decomposition and sampling process DS.

Input: A protocol Π for the problem PC_k.
Output: A rectangle $R = X \times Y$, and $J_A, J_B \subseteq [n]$.

Initialize $v :=$ root of $\Pi, r := 1, X := Y := [n]^n, J_A := J_B := [n]$, bad := FALSE.
while *v is not a leaf node* **do**
  `//Invariant: (1)` $X \times Y \subseteq \Pi_v$`; (2) there exists some` $z_{r-1} \in [n]$ `such that` $\mathsf{pt}_{r-1}(f_A, f_B) = z_{r-1}$ $\forall (f_A, f_B) \in X \times Y$ `(See Lemma 11).`
  Let $u_0 := \Pi(v, 0), u_1 := \Pi(v, 1)$ be the two children of v.
  if owner(v) = Alice **then**
    Partition X into $X = X^0 \cup X^1$ such that $X^b \times Y \subseteq \Pi_{u_b}$ for $b \in \{0, 1\}$.
    Sample $\boldsymbol{b} \in \{0, 1\}$ such that $\mathbf{Pr}[\boldsymbol{b} = b] = |X^b|/|X|$ for $b \in \{0, 1\}$.
    Update $X := X^{\boldsymbol{b}}, v := u_{\boldsymbol{b}}$.
    if owner($u_{\boldsymbol{b}}$) = Bob **then**
      `// A new round.`
      Further Partition X into $X = X^0 \cup X^1$ where $X^b := \{f_A \in X : f_A(z_{r-1}) \bmod 2 = b\}$.
      Sample $\boldsymbol{b'} \in \{0, 1\}$ such that $\mathbf{Pr}[\boldsymbol{b'} = b] = |X^b|/|X|$ for $b \in \{0, 1\}$.
      Update $X := X^{\boldsymbol{b'}}, r := r + 1$.
    Let $X = X^1 \cup \cdots \cup X^m$ be the decomposition of X promised by Lemma 8 with associated sets $I_1, \ldots, I_m \subseteq J_A$.
    `// Invoking Lemma 8 with` $J = J_A, M = n, \gamma = 1 - \frac{0.1}{\log n}$.
    Sample a random element $\boldsymbol{j} \in [m]$ such that $\mathbf{Pr}[\boldsymbol{j} = j] = |X^j|/|X|$ for $j \in [m]$.
    Update $X := X^j, J_A := J_A \setminus I_j$.
    if owner($u_{\boldsymbol{b}}$) = Bob $\wedge\ z_{r-1} \notin J_B$ **then**
      bad := TRUE.
  if owner(v) = Bob **then**
    Partition Y into $Y = Y^0 \cup Y^1$ such that $X \times Y^b \subseteq \Pi_{u_b}$ for $b \in \{0, 1\}$.
    Sample $\boldsymbol{b} \in \{0, 1\}$ such that $\mathbf{Pr}[\boldsymbol{b} = b] = |Y^b|/|Y|$ for $b \in \{0, 1\}$.
    Update $Y := Y^{\boldsymbol{b}}, v := u_{\boldsymbol{b}}$.
    if owner($u_{\boldsymbol{b}}$) = Alice **then**
      Further Partition Y into $Y = Y^0 \cup Y^1$ where $Y^b := \{f_B \in Y : f_B(z_{r-1}) \bmod 2 = b\}$.
      Sample $\boldsymbol{b'} \in \{0, 1\}$ such that $\mathbf{Pr}[\boldsymbol{b'} = b] = |Y^b|/|Y|$ for $b \in \{0, 1\}$.
      Update $Y := Y^{\boldsymbol{b'}}, r := r + 1$.
    Let $Y = Y^1 \cup \cdots \cup Y^m$ be the decomposition of Y promised by Lemma 8 with associated sets $I_1, \ldots, I_m \subseteq J_B$.
    Sample a random element $\boldsymbol{j} \in [m]$ such that $\mathbf{Pr}[\boldsymbol{j} = j] = |Y^j|/|Y|$ for $j \in [m]$.
    Update $Y := Y^j, J_B := J_B \setminus I_j$.
    if owner($u_{\boldsymbol{b}}$) = Alice $\wedge\ z_{r-1} \notin J_A$ **then**
      bad := TRUE.

▶ **Lemma 11** (Loop invariant). *Set* $\gamma \stackrel{\text{def}}{=} 1 - \frac{0.1}{\log n}$. *Then in the running of* $\mathsf{DS}(\Pi)$, *we have the following loop invariants: After each iteration,*

(♢) $X \times Y \subseteq \Pi_v$;

(♣) $\boldsymbol{X}(J_A), \boldsymbol{Y}(J_B)$ *are* γ*-dense;*

(♡) *there exists some* $\alpha_A \in [n]^{\overline{J_A}}, \alpha_B \in [n]^{\overline{J_B}}$ *such that* $x(\overline{J_A}) = \alpha_A, y(\overline{J_B}) = \alpha_B$ *for all* $x \in X, y \in Y$;

(♠) *there exists some* $z_r \in [n]$ *such that* $\mathsf{pt}_r(f_A, f_B) = z_r$ *for all* $(f_A, f_B) \in X \times Y$.

Proof. Item (♢) is true because every time v is updated, $X \times Y$ is updated accordingly to a sub-rectangle of Π_v and updating $X \times Y$ into its sub-rectangles does not violate this condition.

Since we applied density restoring partition at the end of each iteration, Item (♣) and (♡) is guaranteed by Lemma 8 and the way that X, Y, J_A, J_B are updated.

We prove the last item (♠) by induction. Assume that the statement holds after the first $(t-1)$ iterations. WLOG, assume that at the beginning of the t-th iteration, v is owned by Alice. Consider the following two cases.

- Case 1. Not a new round: Line 13 is not executed in the t-th iteration. Since r remains unchanged and we only update R to be a sub-rectangle of itself, the statement still holds.
- Case 2. A new round begins: Line 13 is executed and r is increased by 1. Let ρ denote the value of r before Line 13, then after this iteration, we have $r = \rho + 1$. The induction hypothesis guarantees that there exists some $z_{\rho-1} \in [n]$ such that

  $$\mathsf{pt}_{\rho-1}(f_A, f_B) = z_{\rho-1} \text{ for all} (f_A, f_B) \in X \times Y.$$

  Due to the partition and the update in Line 11 and Line 12, $|\mathrm{supp}(\boldsymbol{X}(z_{\rho-1}))| \le n/2$. Hence, $\boldsymbol{X}(z_{\rho-1})$ cannot be γ-dense as we set $\gamma = 1 - \frac{0.1}{\log n}$. Observe that after the update in Line 17, $\boldsymbol{X}(J_A)$ is γ-dense. Consequently, we must have $z_{\rho-1} \in \overline{J_A}$, and by item (♡), there exists some $z_\rho \in [n]$ such that $f_A(z_{\rho-1}) = z_\rho \ \forall f_A \in X$. By definition, for all $(f_A, f_B) \in X \times Y$,

  $$\mathsf{pt}_\rho(f_A, f_B) = f_A(\mathsf{pt}_{\rho-1}(f_A, f_B)) = f_A(z_{\rho-1}) = z_\rho.$$

  This is exactly the same statement after the t-th iteration (as we have $r = \rho + 1$). ◀

The restricted rectangles in this loop invariant are inspired by the protocols of Nisan and Wigderson [23]. This lemma aims to capture the fact that Alice and Bob cannot get any additional useful information other than coordinate-wise information during their communication.

3.2 Relating Accuracy and Average Fixed Size

From Lemma 11 we know that the coordinates in $\overline{J_A}$ and $\overline{J_B}$ are fixed if we only look at the inputs in $X \times Y$. Intuitively, the advantage of the protocol comes from such fixed coordinates, since the "alive" coordinates J_A, J_B are dense in the sense that $\boldsymbol{X}(J_A), \boldsymbol{Y}(J_B)$ is γ-dense. This intuition is formalized in the following lemma.

▶ **Lemma 12** (Relating accuracy and avarage fixed size). *Let* Π *be a* $(k-1)$*-round deterministic protocol where Alice speaks first. Then*

$$\Pr_{f_A, f_B \leftarrow [n]^n}[\Pi(f_A, f_B) = PC_k(f_A, f_B)] \le \frac{n^{1-\gamma}}{2} + n^{-\gamma} \cdot (k-1) \cdot \mathop{\mathbf{E}}_{(R, J_A, J_B) \leftarrow \mathsf{DS}(\Pi)} \left[|\overline{J_A}| + |\overline{J_B}| \right].$$

The proof of the lemma is by the following two claims. The first claim readily says that conditioned on the flag bad is not raised, Π has little advantage in the rectangle R output by $\mathsf{DS}(\Pi)$. The second claim shows the probability that the flag is raised is bounded in terms of the average fixed size.

▷ Claim 13. If $\mathsf{DS}(\Pi)$ outputs $(R = X \times Y, J_A, J_B)$ and $\mathsf{bad} = \textsc{False}$ in the end, then

$$\Pr_{(f_A,f_B)\leftarrow R}[\Pi(f_A,f_B) = \mathrm{PC}_k(f_A,f_B)] \le \frac{n^{1-\gamma}}{2}.$$

▷ Claim 14. $\Pr_{\mathsf{DS}(\Pi)}[\mathsf{bad} = \textsc{True}] \le n^{-\gamma}\cdot(k-1)\cdot \mathbf{E}_{(R,J_A,J_B)\leftarrow\mathsf{DS}(\Pi)}\left[|\overline{J_A}| + |\overline{J_B}|\right]$.

Next, we first prove Lemma 12 using the above two claims, and the proof of the claims is followed.

Proof of Lemma 12. Note that in the running of $\mathsf{DS}(\Pi)$, we always update R to a randomly chosen rectangle and the probability of each rectangle being chosen is proportional to its size. Consequently,

$$\begin{aligned}
&\Pr_{f_A,f_B\leftarrow[n]^n}[\Pi(f_A,f_B) = \mathrm{PC}_k(f_A,f_B)]\\
=&\Pr_{(R,J_A,J_B)\leftarrow\mathsf{DS}(\Pi),(f_A,f_B)\leftarrow R}[\Pi(f_A,f_B) = \mathrm{PC}_k(f_A,f_B)]\\
\le&\Pr_{\mathsf{DS}(\Pi)}[\mathsf{bad} = \textsc{True}] + \Pr_{(R,J_A,J_B)\leftarrow\mathsf{DS}(\Pi),(f_A,f_B)\leftarrow R}[\Pi(f_A,f_B) = \mathrm{PC}_k(f_A,f_B)\wedge \mathsf{bad} = \textsc{False}]\\
\le&\frac{n^{1-\gamma}}{2} + n^{-\gamma}\cdot(k-1)\cdot\mathop{\mathbf{E}}_{(R,J_A,J_B)\leftarrow\mathsf{DS}(\Pi)}\left[|\overline{J_A}| + |\overline{J_B}|\right].
\end{aligned}$$

where the last step is by Claim 13 and Claim 14. ◀

It remains to prove the two claims.

Proof of Claim 13. WLOG, assume $k-1$ is odd and the protocol always has k round. Let z_{k-1} be the pointer guaranteed by the loop invariant (Lemma 11), i.e., $\mathsf{pt}_{k-1}(f_A,f_B) = z_{k-1}$ for all $(f_A,f_B)\in R$. Since $\mathsf{bad} = \textsc{False}$, we have $z_{k-1}\in J_A$. Again by the loop invariant, $\mathbf{H}_\infty(\boldsymbol{X}(z_{k-1})) \ge \gamma$. Moreover, since R is contained in some leaf node of Π, Π output the same answer in R, say $b^*\in\{0,1\}$. Consequently,

$$\begin{aligned}
\Pr_{(f_A,f_B)\leftarrow R}[\Pi(f_A,f_B) = \mathrm{PC}_k(f_A,f_B)] &= \Pr_{f_A\leftarrow X}[f_A(z_{k-1}) \bmod 2 = b^*]\\
&\le \sum_{\sigma\in[n]:\sigma \bmod 2 = b^*}\Pr_{f_A\leftarrow X}[f_A(z_{k-1}) = \sigma]\\
&\le \frac{n}{2}\cdot n^{-\gamma}.
\end{aligned}$$

◁

Proof of Claim 14. Let $\mathcal{E}_\ell$ denote the event that the flag bad is raised when $r = \ell+1$ (i.e., when the ℓ-th round ends) for the first time. Clearly, $\Pr[\mathsf{bad} = \textsc{True}] = \sum_{\ell=1}^{k-1}\Pr[\mathcal{E}_\ell]$. It suffices to bound each $\Pr[\mathcal{E}_\ell]$.

Assume ℓ is odd, meaning that Alice speaks in the ℓ-th round. Let coin denote the randomness used for the first $(\ell-1)$ rounds. Let $X^{(\ell-1)}, J_A^{(\ell-1)}, J_B^{(\ell-1)}$ be the sets X, J_A, J_B when executing $\mathsf{DS}(\Pi)$ using coin until the ℓ-th round begins. Let $z_{\ell-1}$ be the pointer promised by the invariant. For $\mathcal{E}_\ell$ to happen, we must have $\mathsf{bad} = \textsc{False}$ until the ℓ-th round begins, meaning that $z_{\ell-1}\in J_A^{(\ell-1)}$.

Note that the random variable $\boldsymbol{z}_\ell$ exactly has the same distribution as $\boldsymbol{X}^{(\ell-1)}(z_{\ell-1})$. This is because, in the ℓ-th round (i.e., until r steps to $\ell+1$), we decompose $X^{(\ell-1)}$ into finer sets and update X to be one of them with probability proportional to their size. Therefore,

$$\begin{aligned}\Pr_{\mathsf{coin}'}[\mathcal{E}_t] = \Pr_{\mathsf{coin}'}\left[z_\ell \notin J_B^{(\ell-1)}\right] &= \Pr_{f_A \leftarrow \boldsymbol{X}^{(\ell-1)}}\left[f_A(z_{\ell-1}) \notin J_B^{(\ell-1)}\right] \\ &= \sum_{\sigma \in \overline{J_B^{(\ell-1)}}} \Pr_{f_A \leftarrow \boldsymbol{X}^{(\ell-1)}}\left[f_A(z_{\ell-1}) = \sigma\right] \\ &\le \left|\overline{J_B^{(\ell-1)}}\right| \cdot n^{-\gamma},\end{aligned}$$

where we fix coin and the probability runs over coin', the randomness used afterward; the last inequality holds because $z_{\ell-1} \in J_A^{(\ell-1)}$ and $\boldsymbol{X}^{(\ell-1)}\left(J_A^{(\ell-1)}\right)$ is γ-dense (by Item (♣) in Lemma 11). Averaging over coin, we get

$$\Pr_{\mathsf{DS}(\Pi)}[\mathcal{E}_\ell] \le \mathop{\mathbf{E}}_{\mathsf{coin}}\left[\left|\overline{J_B^{(\ell-1)}}\right|\right] \cdot n^{-\gamma} \le \mathop{\mathbf{E}}_{(R,J_A,J_B)\leftarrow\mathsf{DS}(\Pi)}\left[|\overline{J_B}|\right] \cdot n^{-\gamma},$$

where the second inequality holds because J_B becomes smaller and smaller during the execution.

For even ℓ's, we analogously have $\mathbf{Pr}[\mathcal{E}_\ell] \le \mathbf{E}\left[|\overline{J_A}|\right] \cdot n^{-\gamma}$, and hence the claim follows from union bound. ◁

3.3 Average Fixed Size is Bounded by Communication

Now that the accuracy of a protocol Π is bounded from above by the average fixed size (i.e., $\mathbf{E}_{(R,J_A,J_B)\leftarrow\mathsf{DS}(\Pi)}\left[|\overline{J_A}| + |\overline{J_B}|\right]$), in what follows we show that the average fixed size is at most $O(\mathrm{CC}(\Pi))$. Formally, we prove that

▶ **Lemma 15.** *Let Π be a $(k-1)$-round deterministic protocol where Alice speaks first. Then*

$$\mathop{\mathbf{E}}_{(R,J_A,J_B)\leftarrow\mathsf{DS}(\Pi)}\left[|\overline{J_A}| + |\overline{J_B}|\right] \le \frac{3\mathrm{CC}(\Pi)}{(1-\gamma)\log n}.$$

▶ Remark 16. We shall set $\gamma := 1 - \frac{0.1}{\log n}$ and hence the right-handed side equals $30\mathrm{CC}(\Pi)$.

Proof. We shall prove this lemma by density increment argument. That is, we study the change of the density function

$$\mathbf{D}_\infty(R) \stackrel{\mathsf{def}}{=} \mathbf{D}_\infty(\boldsymbol{X}(J_A)) + \mathbf{D}_\infty(\boldsymbol{Y}(J_B)). \tag{1}$$

in each iteration. Let $\boldsymbol{\phi_t}$ be the value of $\mathbf{D}_\infty(R)$ at the end of the t-th iteration. Assume without loss of generality Alice speaks (i.e., $\mathsf{owner}(v) = \mathsf{Alice}$) in the t-th iteration.

We fix the random coins used for the first $(t-1)$ iterations and consider the updates in the current iteration.

1. First, X is partitioned into $X = X^0 \cup X^1$ according to Π. Then, X is updated to X^b with probability $\frac{|X^b|}{|X|}$. Consequently, $\mathbf{D}_\infty(\boldsymbol{X}(J_A))$ will increase as $|X|$ shrinks, and in expectation (over the random choice of $\boldsymbol{b}$) the increment is

$$\sum_{b\in\{0,1\}} \frac{|X^b|}{|X|} \log\left(\frac{|X|}{|X^b|}\right) \le 1. \tag{2}$$

2. Next, suppose that updating v leads to the switch of the owner, i.e., Line 13 is triggered. Since we also partition X into two parts and update X with probability proportional to the size of each part, the same argument applies. That is, taking expectation over the random choice of $\boldsymbol{b}'$, $\mathbf{D}_\infty(\boldsymbol{X}(J_A))$ increases by at most 1 in expectation.

3. Finally, we further partition X according to Lemma 8. Say X is partitioned into $X = X^1 \cup \cdots \cup X^m$ and let $I_1, \ldots, I_m$ be the index sets promised by Lemma 8; and for all $j \in [m]$ we have

$$\mathbf{D}_\infty(\boldsymbol{X}^j(J_A \setminus I_j)) \le \mathbf{D}_\infty(\boldsymbol{X}(J_A)) - (1-\gamma)\log n|I_j| + \delta_j,$$

where $\delta_j = \log(|X|/\cup_{v \ge j} X^v)$. With probability $p_j \stackrel{\text{def}}{=} |X^j|/|X|$, we update $X := X^j$ and $J_A := J_A \setminus I_j$. Therefore, taking expectation over the random choice of $\boldsymbol{j}$, the density function will decrease by

$$\mathbf{D}_\infty(\boldsymbol{X}(J_A)) - \mathop{\mathbf{E}}_{j \leftarrow \boldsymbol{j}}\left[\mathbf{D}_\infty(\boldsymbol{X}^j(J_A \setminus I_j))\right] \ge \mathop{\mathbf{E}}_{j \leftarrow \boldsymbol{j}}\left[(1-\gamma)\log n \cdot |I_j| - \delta_j\right]. \tag{3}$$

Note that $\delta_j \stackrel{\text{def}}{=} \log \frac{1}{\sum_{v \ge j} p_v}$ and thus

$$\mathop{\mathbf{E}}_{j \leftarrow \boldsymbol{j}}\left[\delta_j\right] = \sum_{j=1}^{m} p_j \log \frac{1}{\sum_{v \ge j} p_j} \le \int_0^1 \log \frac{1}{1-x}\mathrm{d}x \le 1. \tag{4}$$

Let $\mathcal{F}_{t-1}$ be the σ-algebra generated by the random coins used for the first $(t-1)$ iterations. Let $\boldsymbol{\beta_t}$ be the increment of $|\overline{J_A}|$ and $|\overline{J_B}|$ in the t-th iteration. Observe that $\boldsymbol{\beta_t} = |I_j|$ by definition. By Equation (3) and Equation (4), taking expectation over random choice of $\boldsymbol{j}$, $\mathbf{D}_\infty(\boldsymbol{X}(J_A))$ decrease by at least $(1-\gamma)\log n \cdot \mathbf{E}\left[\boldsymbol{\beta_t} \mid \mathcal{F}_{t-1}\right] - 1$ due to the density restoring partition. Then

$$\mathbf{E}\left[\boldsymbol{\phi_t} - \boldsymbol{\phi}_{t-1}\right] = \mathbf{E}\left[\mathbf{E}\left[\boldsymbol{\phi_t} - \boldsymbol{\phi}_{t-1} \mid \mathcal{F}_{t-1}\right]\right] \le \mathbf{E}\left[1 + \boldsymbol{\eta_t} - ((1-\gamma)\log n \cdot \boldsymbol{\beta_t} - 1)\right], \tag{5}$$

where $\boldsymbol{\eta_t} \stackrel{\text{def}}{=} \mathbb{1}[\text{owner switches in the } t\text{-th iteration}]$.

Write $c \stackrel{\text{def}}{=} \mathrm{CC}(\Pi)$ and assume we always have c iterations. [1] In the beginning, $\boldsymbol{\phi}_0 = \mathbf{D}_\infty([n]^n \times [n]^n) = 0$. Since the density function is always non-negative by definition, we have $\boldsymbol{\phi}_c \ge 0$ and thus $\mathbf{E}\left[\boldsymbol{\phi}_c - \boldsymbol{\phi}_0\right] \ge 0$. On the other hand, by telescoping,

$$\mathbf{E}\left[\boldsymbol{\phi}_c - \boldsymbol{\phi}_0\right] = \sum_{t=1}^{c} \mathbf{E}\left[\boldsymbol{\phi_t} - \boldsymbol{\phi}_{t-1}\right] \le 2c + \sum_{t=1}^{c} \mathbf{E}\left[\boldsymbol{\eta}_t - (1-\gamma)\log n \cdot \boldsymbol{\beta_t}\right],$$

where the inequality follows from Equation (5). Observe that $\sum_{t=1}^{c} \boldsymbol{\eta}_t$ is at most k and $\sum_{t=1}^{c} \boldsymbol{\beta}_t = |\overline{J_A}| + |\overline{J_B}|$ by definition. We conclude that

$$\mathbf{E}\left[|\overline{J_A}| + |\overline{J_B}|\right] = \mathbf{E}\left[\sum_{t=1}^{c} \boldsymbol{\beta}_t\right] \le \frac{2c+k}{(1-\gamma)\log n} \le \frac{3c}{(1-\gamma)\log n},$$

as desired. ◀

Proving the main theorem

Now our main theorem easily follows from the two lemmas.

[1] Namely, Π communicates c bits on all inputs.

Proof of Theorem 2. Set $\gamma \stackrel{\text{def}}{=} 1 - \frac{0.1}{\log n}$. By Lemma 15 and Lemma 12, we get

$$\begin{aligned}\texttt{Accuracy}(\Pi) &\stackrel{\text{def}}{=} \mathop{\mathbf{Pr}}_{f_A, f_B \leftarrow [n]^n}[\Pi(f_A, f_B) = \mathrm{PC}_k(f_A, f_B)] \\ &\le \frac{n^{1-\gamma}}{2} + n^{-\gamma} \cdot (k-1) \cdot \frac{3\mathrm{CC}(\Pi)}{(1-\gamma)\log n} \\ &\le 0.54 + \frac{1.08(k-1)}{n} \cdot 30\mathrm{CC}(\Pi),\end{aligned}$$

where we use $\frac{n^{1-\gamma}}{2} \le 0.54, n^{-\gamma} \le \frac{1.08}{n}$. Since we assumed $\texttt{Accuracy}(\Pi) \ge 2/3$, we conclude that

$$\mathrm{CC}(\Pi) \ge \frac{2/3 - 0.54}{1.08 \cdot 30} \cdot \frac{n}{k-1} > 0.0039 \cdot \frac{n}{k-1} = \Omega(n/k).$$

We also trivially have $\mathrm{CC}(\Pi) \ge k - 1$ as Π has $(k-1)$ rounds; putting it together we conclude that $\mathrm{CC}(\Pi) = \Omega(n/k + k)$. ◀

References

1. Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Polynomial pass lower bounds for graph streaming algorithms. In *Proceedings of the 51st Annual ACM SIGACT Symposium on theory of computing*, pages 265–276, 2019. doi:10.1145/3313276.3316361.
2. Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 67–76, 2010. doi:10.1145/1806689.1806701.
3. Joakim Blikstad, Jan Van Den Brand, Yuval Efron, Sagnik Mukhopadhyay, and Danupon Nanongkai. Nearly optimal communication and query complexity of bipartite matching. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1174–1185. IEEE, 2022. doi:10.1109/FOCS54457.2022.00113.
4. Mark Braverman, Anup Rao, Omri Weinstein, and Amir Yehudayoff. Direct products in communication complexity. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 746–755. IEEE, 2013. doi:10.1109/FOCS.2013.85.
5. Clément L Canonne and Tom Gur. An adaptivity hierarchy theorem for property testing. *computational complexity*, 27:671–716, 2018. doi:10.1007/S00037-018-0168-4.
6. Arkadev Chattopadhyay, Yuval Filmus, Sajin Koroth, Or Meir, and Toniann Pitassi. Query-to-communication lifting for bpp using inner product. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.35.
7. Xi Chen, Christos Papadimitriou, and Binghui Peng. Memory bounds for continual learning. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 519–530. IEEE, 2022. doi:10.1109/FOCS54457.2022.00056.
8. Carsten Damm, Stasys Jukna, and Jiří Sgall. Some bounds on multiparty communication complexity of pointer jumping. *Computational Complexity*, 7:109–127, 1998. doi:10.1007/PL00001595.
9. Pavol Duris, Zvi Galil, and Georg Schnitger. Lower bounds on communication complexity. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 81–91, 1984. doi:10.1145/800057.808668.
10. Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. Graph distances in the data-stream model. *SIAM Journal on Computing*, 38(5):1709–1727, 2009. doi:10.1137/070683155.
11. Mikael Goldmann and Johan Håstad. A simple lower bound for monotone clique using a communication game. *Information Processing Letters*, 41(4):221–226, 1992. doi:10.1016/0020-0190(92)90184-W.

12 Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1077–1088. IEEE, 2015. doi:10.1109/FOCS.2015.70.

13 Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for bpp. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 132–143, 2017. doi:10.1109/FOCS.2017.21.

14 Sudipto Guha and Andrew McGregor. Lower bounds for quantile estimation in random-order and multi-pass streaming. In *Automata, Languages and Programming: 34th International Colloquium, ICALP 2007, Wrocław, Poland, July 9-13, 2007. Proceedings 34*, pages 704–715. Springer, 2007. doi:10.1007/978-3-540-73420-8_61.

15 Venkatesan Guruswami and Krzysztof Onak. Superlinear lower bounds for multipass graph processing. *Algorithmica*, 76:654–683, 2016. doi:10.1007/S00453-016-0138-7.

16 Matthew Joseph, Jieming Mao, and Aaron Roth. Exponential separations in local differential privacy. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 515–527. SIAM, 2020. doi:10.1137/1.9781611975994.31.

17 Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 767–775, 2002. doi:10.1145/509907.510017.

18 Hartmut Klauck. On quantum and probabilistic communication: Las vegas and one-way protocols. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 644–651, 2000. doi:10.1145/335305.335396.

19 Hartmut Klauck, Ashwin Nayak, Amnon Ta-Shma, and David Zuckerman. Interaction in quantum communication. *IEEE Transactions on Information Theory*, 53(6):1970–1982, 2007. doi:10.1109/TIT.2007.896888.

20 Maria Klawe, Wolfgang J Paul, Nicholas Pippenger, and Mihalis Yannakakis. On monotone formulae with restricted depth. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 480–487, 1984.

21 Shachar Lovett, Raghu Meka, Ian Mertz, Toniann Pitassi, and Jiapeng Zhang. Lifting with sunflowers. *Leibniz international proceedings in informatics*, 215, 2022. doi:10.4230/LIPICS.ITCS.2022.104.

22 Danupon Nanongkai, Atish Das Sarma, and Gopal Pandurangan. A tight unconditional lower bound on distributed randomwalk computation. In *Proceedings of the 30th annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, pages 257–266, 2011. doi:10.1145/1993806.1993853.

23 Noam Nisan and Avi Widgerson. Rounds in communication complexity revisited. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 419–429, 1991. doi:10.1145/103418.103463.

24 Binghui Peng, Srini Narayanan, and Christos Papadimitriou. On limitations of the transformer architecture. *arXiv preprint*, 2024. doi:10.48550/arXiv.2402.08164.

25 Stephen J Ponzio, Jaikumar Radhakrishnan, and Srinivasan Venkatesh. The communication complexity of pointer chasing. *Journal of Computer and System Sciences*, 62(2):323–355, 2001. doi:10.1006/JCSS.2000.1731.

26 Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020.

27 Ran Raz and Pierre McKenzie. Separation of the monotone nc hierarchy. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 234–243. IEEE, 1997. doi:10.1109/SFCS.1997.646112.

28 Guangxu Yang and Jiapeng Zhang. Communication lower bounds for collision problems via density increment arguments. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 630–639, 2024. doi:10.1145/3618260.3649607.

29 Amir Yehudayoff. Pointer chasing via triangular discrimination. *Combinatorics, Probability and Computing*, 29(4):485–494, 2020. doi:10.1017/S0963548320000085.

A Appendix

The following lemma and proof are from Lemma 5 in [13].

▶ **Lemma 17** (Lemma 8 restated). *Let $\gamma \in (0,1)$. Let X be a subset of $[n]^M$ and $J \subseteq [M]$. Suppose that there exists an $\beta \in [n]^{\overline{J}}$ such that $\forall x \in X, x(\overline{J}) = \beta$. Then, there exists a partition $X = X^1 \cup X^2 \cup \cdots \cup X^r$ and every X^i is associated with a set $I_i \subseteq J$ and a value $\alpha_i \in \{0,1\}^{I_i}$ that satisfy the following properties.*

1. $\forall x \in X^i, x(I_i) = \alpha_i$;
2. $\boldsymbol{X}^i(J \setminus I_i)$ *is γ-dense;*
3. $\mathbf{D}_\infty\left(\boldsymbol{X}^i(J \setminus I_i)\right) \le \mathbf{D}_\infty\left(\boldsymbol{X}(J)\right) - (1-\gamma)\log n \cdot |I_i| + \delta_i$, *where* $\delta_i \stackrel{\text{def}}{=} \log(|X|/|\cup_{j \ge i} X^j|)$.

Proof. We prove it by a greedy algorithm as follows.

Algorithm 2 Greedy Algorithm.

Input: $X \subseteq [n]^M$
Output: A partition $X = X^1 \cup X^2 \cup \cdots \cup X^m$
Initialize $i := 1$.
while $X \neq \emptyset$ **do**
  Let $I \subseteq J$ be a maximal subset (possibly $I = \emptyset$) such that $\mathbf{H}_\infty(\boldsymbol{X}(I)) < \gamma|I|\log n$ and let $\alpha_i \in [n]^I$ be a witness of this fact, i.e., $\mathbf{Pr}[\boldsymbol{X}(I) = \alpha_i] > n^{-\gamma|I|}$.
  $X^i := \{x \in X : x(I) = \alpha_i\}$ and $I_i := I$.
  Update $X := X \setminus X^i$, $J := J \setminus I_i$, and $i := i + 1$.

Item 1 is guaranteed by the construction of X^i and I_i.

We prove Item 2 by contradiction. Assume towards contradiction that $\boldsymbol{X}^i(J \setminus I_i)$ is not γ-dense for some i. By definition, there is a nonempty set $K \subseteq J \setminus I_i$ and $\beta \in [n]^K$ violating the min-entropy condition, namely, $\mathbf{Pr}\left[\boldsymbol{X}(K) = \beta\right] > n^{-\gamma|K|}$. Write $X^{\ge i} \stackrel{\text{def}}{=} \cup_{j \ge i} X^i$. Then

$$\mathbf{Pr}\left[\boldsymbol{X}^{\ge i}(I_i \cup K) = (\alpha_i, \beta)\right] = \mathbf{Pr}\left[\boldsymbol{X}^{\ge i}(I_i) = \alpha_i\right] \cdot \mathbf{Pr}\left[\boldsymbol{X}^i(K) = \beta\right] > n^{-\gamma|I_i|} \cdot n^{-\gamma|K|} = n^{-\gamma|I_i \cup K|},$$

where the first equality holds as $(\boldsymbol{X}^{\ge i} | \boldsymbol{X}^{\ge i}(I_i) = \alpha_i) = \boldsymbol{X}^i$. However, this means at moment that I_i is chosen, the set $I_i \cup K \subseteq J$ also violates the min-entropy condition (witnessed by (α_i, β)), contradicting the maximality of I_i.

Finally, Item 3 is proved by straightforward calculation:

$$\begin{aligned}
\mathbf{D}_\infty(\boldsymbol{X}^i(J \setminus I_i)) &= |J \setminus I_i|\log n - \log|X^i| \\
&\le (|J|\log n - |I_i|\log n) - \log\left(\left|X^{\ge i}\right| \cdot n^{-\gamma|I_i|}\right) \\
&= (|J|\log n - \log|X|) - (1-\gamma)|I_i| \cdot \log n + \log\left(\frac{|X|}{|X^{\ge i}|}\right) \\
&= \mathbf{D}_\infty\Big(\boldsymbol{X}(J)\Big) - (1-\gamma)|I_i|\log n + \delta_i.
\end{aligned}$$

◀

A Quantum Unique Games Conjecture

Hamoon Mousavi
Simons Institute for Theoretical Computer Science, University of California, Berkeley, CA, USA

Taro Spirig
QMATH, Department of Mathematical Sciences, University of Copenhagen, Denmark

Abstract

After the NP-hardness of computational problems such as 3SAT and MaxCut was established, a natural next step was to explore whether these problems remain hard to approximate. While the quantum nonlocal games extensions of some of these problems are known to be hard – indeed undecidable – their inapproximability remains largely unresolved. In this work, we introduce definitions for the quantum extensions of Label-Cover and Unique-Label-Cover. We show that these problems play a similarly crucial role in studying the inapproximability of quantum constraint satisfaction problems as they do in the classical setting.

2012 ACM Subject Classification Theory of computation → Quantum complexity theory; Theory of computation → Quantum complexity theory

Keywords and phrases hardness of approximation, quantum computing, noncommutative constraint satisfaction problems, Fourier analysis

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.76

Related Version *Full Version*: `https://arxiv.org/abs/2409.20028` [38]

Funding *Hamoon Mousavi*: Supported by a Simons-CIQC postdoctoral fellowship through NSF QLCI Grant No. 2016245.
Taro Spirig: Supported by the European Union under the Grant Agreement No 101078107, QInteract and VILLUM FONDEN via Villum Young Investigator grant (No 37532) and the QMATH Centre of Excellence (Grant No 10059).

Acknowledgements We thank Henry Yuen for valuable discussions on the topics explored in this paper. We thank Eric Culf and Michael Chapman for helpful discussions.

1 Introduction

1.1 Motivations

The Label-Cover problem [1, 8, 16], a well-known constraint satisfaction problem (CSP), has been central in the study of hardness of approximation. According to the PCP theorem [2, 3, 4, 5, 8, 15, 19, 45], it is NP-hard to approximate Label-Cover within any additive constant. This result, combined with gap-preserving reductions from Label-Cover, has led to optimal inapproximability results for several other key problems, including 3XOR and 3SAT, due largely to the work of Håstad [24, 25].

An instance of Label-Cover consists of a bipartite graph with left vertex set U and right vertex set V, a large label set (also called an alphabet) $\mathcal{L}$, and maps $\pi_{u,v} : \mathcal{L} \to \mathcal{L}$ associated with each edge (u, v). These maps are usually called projections. A *labelling* $a : U \cup V \to \mathcal{L}$ is said to satisfy an edge (u, v) if it respects the associated projection map, meaning if $\pi_{u,v}(a(u)) = a(v)$. The *value of the instance* is the maximum fraction of satisfied edges over all possible labelings.

© Hamoon Mousavi and Taro Spirig;
licensed under Creative Commons License CC-BY 4.0

In quantum information theory, the quantum extension of Label-Cover, known as *entangled projection games* [14], is a special case of *nonlocal games* [10, 17].[1] The primary distinction lies in the definition of labeling. A *quantum labeling* generalizes probabilistic labeling. In classical probabilistic labeling, each vertex is assigned a probability distribution over labels. In quantum labeling, each edge is assigned a distribution over pairs of labels – one for each vertex incident on the edge.

More formally, a quantum labeling assigns quantum measurements to each vertex, with the measurement outcomes corresponding to labels. When both vertices of an edge are measured, the resulting probability distribution determines the likelihood that the edge is satisfied. The *quantum value* of the instance is the maximum expected fraction of satisfied edges over all possible quantum labelings. A formal definition is given in Section 3.1 of the full version [38].

In the quantum setting, the role of the PCP theorem is played by the $\mathrm{MIP}^* = \mathrm{RE}$ theorem of Ji et al. [29]: it is RE-hard to approximate the quantum value of Label-Cover within any additive constant.[2] Now, as with the classical case, one might expect to derive optimal inapproximability results for other quantum constraint satisfaction problems. O'Donnell and Yuen [41], for instance, extended Håstad's results to the optimal inapproximability of quantum 3XOR. So far, the classical and quantum theories of inapproximability have followed parallel trajectories.

In the classical development, after the initial success of Håstad, researchers discovered some of the limitations of Label-Cover. To address these, Khot [32] introduced a modification to the Label-Cover problem, imposing a uniqueness condition on the projection maps. This modified problem is known as the Unique-Label-Cover and it remains an open question whether it is hard to approximate. However, assuming that Unique-Label-Cover is hard to approximate, researchers showed optimal inapproximability results for a variety of other problems, including MaxCut [34]. This assumption is known as the Unique Games Conjecture (UGC) [32].

A striking divergence arises between the classical and quantum theories of inapproximability with the introduction of Unique-Label-Cover. Kempe et al. [31] showed that the quantum value of Unique-Label-Cover is efficiently approximable. This implies that a *direct* quantum analogue of the UGC does not hold. As a result, none of the classical consequences of the UGC can be straightforwardly extended to the quantum domain. This presents a significant challenge to the development of a comprehensive quantum theory for hardness of approximation.

For instance, understanding the complexity of the anti-ferromagnetic Heisenberg model – a quantum generalization of MaxCut and a problem of great significance in physics and quantum information – remains an open question. While this problem is known to be QMA-hard, the complexity of approximating it is still unresolved.[3] Hwang et al. [26] made significant progress on this question, but their results rely on the classical UGC, which limits them to showing NP-hardness rather than QMA-hardness.

Finding the right candidate for a quantum extension of UGC is key for advancing the theory of inapproximability in the quantum setting. This work proposes such a candidate.

---

[1] We primarily adopt the CSP terminology, while referencing certain naming conventions from the nonlocal games literature for the convenience of readers familiar with that field. See Section 1.2 for a discussion of our terminology.

[2] Recall that RE is the class of recursively enumerable languages. Stating that a problem is RE-hard is another way of saying that it is undecidable.

[3] Recall that QMA is the quantum analogue of NP.

We begin by noting that Kempe et al.'s algorithm for approximating the quantum value of Unique-Label-Cover applies only to some of many possible quantum values that are defined in the literature on nonlocal games. Examples of quantum values include, but are not limited to, the tensor-product, commuting-operator, synchronous, and oracularizable-synchronous values. Label-Cover remains hard across all these models [29, 47, 48], whereas Unique-Label-Cover is efficiently approximable in only some of them [31].

We choose a model where Unique-Label-Cover might still be hard to approximate. This choice forms the basis of our *quantum Unique Games Conjecture* (qUGC). We give an informal overview in Section 1.3. Formal definitions can be found in Section 3.2 of the full version [38].

We then show how classical reductions from Unique-Label-Cover can be "quantized" to prove inapproximability results for quantum CSPs. We illustrate this approach with 2-Lin [32] and MaxCut [34], and we believe the same tools can be applied to Raghavendra's seminal work on CSPs [44]. An informal overview is given in Section 1.4. The proof ideas are sketched in Section 1.5. Detailed statements and proofs are provided in Sections 4 and 5 of the full version [38].

We conclude this section with a few remarks.

Relationship between UGC and qUGC. The cases of 3XOR, 2-Lin, and MaxCut could be instances of a broader principle: any reduction from Label-Cover (or its variants such as Unique-Label-Cover, Smooth-Label-Cover, and 2-to-2 Label-Cover) might be adaptable to the quantum setting. If true, this would imply that the classical UGC implies qUGC. We explore this possibility and related open questions in Section 1.6.

Variants of qUGC. Section 1.3 and Section 3.2 in the full version [38] present several variants of the quantum Unique Games Conjecture. These variants may help in proving hardness for other models, such as the anti-ferromagnetic Heisenberg model or CSPs investigated in [12]. These topics are further detailed as Problems 4, 7, and 8 in Section 1.6.

Alternative Approaches to UGC. Some of the variants of quantum UGC proposed in this paper may be easier to prove hardness for than the classical UGC itself. Proving hardness for any of these variants could strengthen confidence in the UGC. Conversely, some of our UGC variants may be easier to disprove, and doing so could offer insights into disproving UGC. Figure 5, illustrates the reductions between variants of Unique-Label-Cover.

Universality. Label-Cover is exhibiting a remarkable *universality*, remaining hard across all the different quantum and classical models, see Figure 4. In contrast, the brittleness of Unique-Label-Cover aptly mirrors the difficulty in resolving the UGC. From another perspective, however, the constraints encountered in the search for a quantum analogue of the UGC could serve as a guiding principle in identifying the most appropriate model for quantum value from a computational standpoint.

This paper builds on our previous work on noncommutative CSPs [12], exploring how noncommutativity reshapes the landscape of CSPs. One of our motivations is the direct applications of this research to quantum information. We are also equally interested in whether a broader perspective on CSPs might uncover unifying principles that could provide new insights into classical CSPs. Our remarks on alternative approaches to UGC and universality are examples of the insights we are pursuing.

1.2 Quantum Value

We illustrate the concept of *quantum value* of a CSP instance using the example of MaxCut. For formal definitions refer to Section 3 in the full version [38]. Before proceeding, we briefly outline the connection to *nonlocal games*; however, no prior knowledge in quantum information is required to follow this work.

Nonlocal Games. The types of values we introduce for CSPs correspond directly to the values for *nonlocal games* previously studied in quantum information. Every CSP instance can be viewed as a nonlocal game and vice versa and the correspondence between their values is straightforward. This relationship is detailed in Section 10.2 of [12]. The physical, mathematical, and computational considerations for the definitions of the various types of values are extensively explored in the literature on nonlocal games [6, 9, 10, 20, 23, 29, 30, 35, 37, 42, 43, 46, 47, 48, 50, 51]. For readers familiar with this area, Table 1 summarizes the correspondences relevant to this paper. Please also refer to the discussion on previous work at the end of this section for our choice of terminology.

Table 1 Correspondence between types of values for CSPs and nonlocal games. Formal definitions for the values in the left column are given in Section 3 of the full version [38].

CSP terminology	Nonlocal game terminology
classical	classical synchronous
noncommutative [12]	quantum synchronous [43, 23, 35, 37]
quantum	oracularizable quantum synchronous [23, 29, 37]

MaxCut. The objective of MaxCut is to partition the vertices of a given graph $G = (V, E)$ into two subsets, such that the number of edges crossing the partition is maximized (as illustrated in Figure 1).

Classical Value. A (classical) partition can be represented as an assignment $x : V \to \{-1, +1\}$. The value of this partition, meaning the number of edges that cross the partition, is given by $\sum_{(i,j)\in E} \frac{1-x_i x_j}{2}$. The (classical) value of the instance, denoted $\omega_c(G)$, is the maximum partition value over all possible (classical) partitions:

$$\begin{aligned} &\text{maximize:} && \sum_{(i,j)\in E} \frac{1 - x_i x_j}{2} \\ &\text{subject to:} && x_i \in \{-1, +1\}. \end{aligned} \tag{1}$$

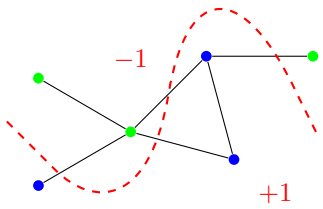


Figure 1 An instance of Max-Cut and an example of a partition (cut).

Noncommutative Value. A *noncommutative partition* assigns an observable to each vertex, i.e. $X : V \to \mathrm{Obs}(\mathcal{H})$, where $\mathcal{H}$ is some finite-dimensional Hilbert space. Observables are unitary operators with eigenvalues in the set $\{-1, +1\}$ (refer to Section 2.2 in the full version [38] for background on quantum measurements and observables). If we substitute observables for ± 1 in (1), we obtain the noncommutative value of the instance, denoted $\omega_{\mathrm{nc}}(G)$:

$$\begin{aligned} \text{maximize:} \quad & \sum_{(i,j)\in E} \frac{1 - \mathrm{tr}(X_i X_j)}{2} \\ \text{subject to:} \quad & X_i \in \mathrm{Obs}(\mathcal{H}), \end{aligned} \tag{2}$$

where tr is the dimension-normalized trace on $\mathcal{H}$. Crucially, the optimization is over all finite-dimensional $\mathcal{H}$. When $\mathcal{H}$ is one-dimensional, we recover the classical value, therefore $\omega_{\mathrm{nc}}(G) \geq \omega_c(G)$.

The noncommutative value has been the focus of our previous work on CSPs [12]. The main focus of this paper, however, is the *quantum value* of CSPs, which we define next.

Quantum Value. A *quantum partition* lies somewhere in between a classical and noncommutative partition: A noncommutative partition X is *quantum* if X_i and X_j commute whenever (i, j) is an edge. In quantum mechanics, commuting observables can be simultaneously measured, emphasizing the physical motivation for this additional commutation relation.

The *quantum value*, denoted $\omega_q(G)$, is therefore obtained from (2) by imposing the extra commutation relations:

$$\begin{aligned} \text{maximize:} \quad & \sum_{(i,j)\in E} \frac{1 - \mathrm{tr}(X_i X_j)}{2} \\ \text{subject to:} \quad & X_i \in \mathrm{Obs}(\mathcal{H}), \\ & X_i X_j = X_j X_i \text{ for all } (i,j) \in E. \end{aligned} \tag{3}$$

Since every classical assignment is a quantum assignment and every quantum assignment is a noncommutative assignment, we obtain the chain of inequalities $\omega_{\mathrm{nc}}(G) \geq \omega_q(G) \geq \omega_c(G)$.

Using the same approach outlined here, we can define the classical and quantum values for all CSPs (refer to Section 3 in the full version [38] for the relevant definitions). However, the noncommutative value is defined only for CSPs where each constraint involves only two variables.[4]

We assume $\omega_c, \omega_q, \omega_{\mathrm{nc}}$ are normalized appropriately so that they are between 0 and 1. So for example in (1),(2), and (3), the normalization amounts to dividing the objective function by the number of edges.

Remark. It is important to note that the commutativity requirement in a quantum assignment does not imply that all observables in the assignment commute with each other, nor does it mean that all observables can be diagonalized in the same basis. Commutativity is not a transitive property: every operator commutes with the identity operator, but not every

[4] This restriction is related to a similar issue for synchronous value of nonlocal games: with current definitions in the literature, the quantum synchronous value of a three-player nonlocal game reduces to the classical synchronous value. See also the remark after Definition 13 in the full version [38].

pair of operators commute with each other. For example, in MaxCut, if two vertices are at a distance of two or more, their observables in a quantum assignment need not commute.

Previous Work. The concept of noncommutative constraint satisfaction problems studied in this work traces its roots to the binary constraint system games introduced by Cleve and Mittal [11]. To the best of our knowledge, the first work to introduce a concept closely related to our notion of quantum assignment – referred to as oracularizable strategies for entangled nonlocal games – is by Ito et al. [27]. Oracularization is also a widely used technique in the study of classical nonlocal games (see references in [27]). Helton et al. [23] later examined oracularizable strategies from a mathematical perspective, employing the term locally commuting algebras (see Sections 7 and 8 of their paper).

Our definition of quantum assignment is the direct translation (for CSPs) of the *oracularizable quantum synchronous strategies* in [37]. The definition in [37] itself dates back to the breakthrough results [40, 29]. In this work, we refer to these assignments as *quantum* because the requirement of *commutativity* stems from *simultaneous measurability in quantum mechanics*. In contrast, we refer to assignments as *noncommutative* when there is no commutation requirement. This terminology originates from our earlier work on noncommutative CSPs [12].[5]

1.3 Quantum Unique Games Conjecture

Before moving on to our conjecture, we first briefly review decision problems and reductions between them. Readers familiar with the terminology can skip the next section.

1.3.1 Reductions

In the context of hardness of approximation, it is often easier to work with a version of a computational problem known as the *decision problem with a promise*. Let $\mathcal{P}$ be any computational problem for which $\omega_c, \omega_q, \omega_{\mathrm{nc}}$ are defined. For example $\mathcal{P}$ could be MaxCut, Label-Cover, or Unique-Label-Cover. For every $s, t \in \{c, q, \mathrm{nc}\}$ and $\zeta, \delta > 0$, the *decision problem* $\mathcal{P}_{s,t}(1-\zeta, \delta)$ is defined as follows: given an instance $\phi \in \mathcal{P}$ that is promised to be either

- $\omega_s(\phi) \geq 1 - \zeta$, called a "yes" instance, or
- $\omega_t(\phi) < \delta$, called a "no" instance,

decide whether the instance is a "yes" or "no" instance. When $s = t$, we simply write $\mathcal{P}_s(1-\zeta, \delta)$ to denote $\mathcal{P}_{s,s}(1-\zeta, \delta)$.

A *reduction* from $\mathcal{P}_{s,t}(\gamma, \delta)$ to $\mathcal{P}'_{s',t'}(\gamma', \delta')$ is any efficient map $\mathcal{P} \to \mathcal{P}'$ that is

- complete: it maps "yes" instances to "yes" instances, and
- sound: it maps "no" instances to "no" instances.

As a simple example, for every choice of $\zeta, \delta > 0$, the decision problem $\mathcal{P}_{q,\mathrm{nc}}(1-\zeta, \delta)$ reduces to $\mathcal{P}_{\mathrm{nc},\mathrm{nc}}(1-\zeta, \delta)$ via the identity map. Soundness is trivial, and completeness holds because $\omega_{\mathrm{nc}}(\phi) \geq \omega_q(\phi)$ for all instances ϕ. This is denoted by $\mathcal{P}_{q,\mathrm{nc}} \longrightarrow \mathcal{P}_{\mathrm{nc},\mathrm{nc}}$ (we often drop the completeness and soundness parameters in our informal discussions in this introduction). There are several other trivial reductions between decision versions of $\mathcal{P}$, and they are illustrated in Figure 2.

[5] Our decision to use *quantum synchronous strategies*, both here and in [12], is driven by the syntactical simplicity they provide compared to tensor-product strategies. While we believe that all of our results can be extended to the tensor-product model, we leave a more detailed exploration of this for future work.

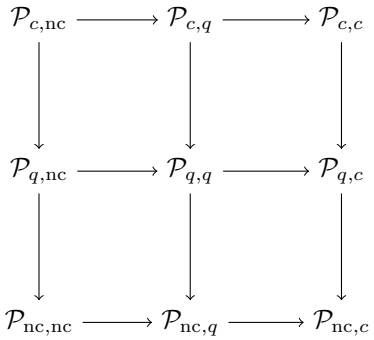


Figure 2 Trivial Reductions: if a problem in this diagram is hard (for some complexity class), then every problem reachable from it is at least as hard.

Lastly, we use the abbreviations LC for Label-Cover and ULC for Unique-Label-Cover.

1.3.2 Label-Cover

The PCP theorem states that the classical value of Label-Cover is NP-hard to approximate, or more precisely:

▶ **Theorem 1** (PCP Theorem). *For every $\delta > 0$, there is a large enough alphabet over which* $\mathrm{LC}_c(1, \delta)$ *is* NP*-hard.*

Since we have the trivial reductions $\mathrm{LC}_{c,c} \longrightarrow \mathrm{LC}_{q,c} \longrightarrow \mathrm{LC}_{\mathrm{nc},c}$, we conclude that both $\mathrm{LC}_{q,c}$ and $\mathrm{LC}_{\mathrm{nc},c}$ are at least NP-hard. This is illustrated in Figure 3.

What can we say about the other variants of Label-Cover in Figure 3? The noncommutative analogue of the PCP theorem is the $\mathrm{MIP}^* = \mathrm{RE}$ theorem. This theorem states that the noncommutative value of Label-Cover is RE-hard to approximate. In fact [29] proves a stronger theorem:

▶ **Theorem 2** ($\mathrm{MIP}^* = \mathrm{RE}$, informal). *For every $\delta > 0$, the decision problem* $\mathrm{LC}_{q,\mathrm{nc}}(1, \delta)$ *is* RE*-hard.*

This immediately implies that the problems in the last two rows in Figure 3 are RE-hard. Our full knowledge of the complexity landscape of Label-Cover is summarized in Figure 4.

1.3.3 Unique-Label-Cover

A Unique-Label-Cover instance is a special case of Label-Cover instance where every projection map is a bijection (i.e. a permutation), see Section 3.2 in the full version [38] for further details. The well-known Unique Games Conjecture asserts that it is hard to approximate the classical value of Unique-Label-Cover:

▶ **Conjecture 3** (UGC [32], informal). *For all $\zeta, \delta > 0$, the decision problem* $\mathrm{ULC}_c(1 - \zeta, \delta)$ *is* NP*-hard.*

Kempe et al. [31] provided an approximation algorithm for the noncommutative value of Unique-Label-Cover (see Theorem 19 in the full version [38] for details), but this result does not address the quantum value. Thus, we propose the following quantum analogue of the Unique Games Conjecture (see Conjecture 20 in the full version [38] for the precise statement):

▶ **Conjecture 4** (qUGC, informal). *For all $\zeta, \delta > 0$, the decision problem* $\mathrm{ULC}_q(1 - \zeta, \delta)$ *is* RE*-hard.*

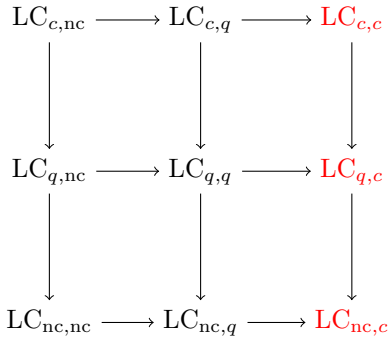


Figure 3 Implication of the PCP theorem: problems in the third column are NP-hard.

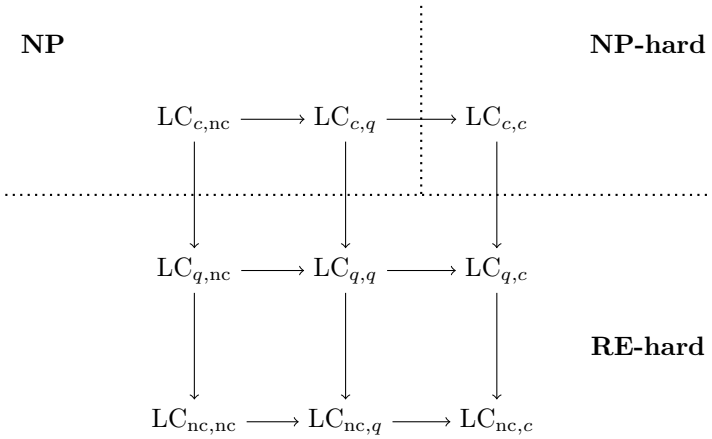


Figure 4 The complexity landscape of Label-Cover.

Assuming both UGC and qUGC, along with the algorithm from Kempe et al., the complexity landscape of Unique-Label-Cover can be summarized as shown in Figure 5.

Assuming that any one of $\mathrm{ULC}_{q,q}, \mathrm{ULC}_{q,c}, \mathrm{ULC}_{\mathrm{nc},q}, \mathrm{ULC}_{\mathrm{nc},c}$ is RE-hard constitutes a variant of the qUGC. The strongest among these variants is the conjecture concerning $\mathrm{ULC}_{q,q}$. As discussed in Section 3.2 in the full version [38], there are many (indeed, infinitely many) variants of qUGC. When we refer to qUGC without additional qualifiers, we are specifically referring to the conjecture stated above.

Given the reductions shown in Figure 5, it is likely easier to prove hardness for $\mathrm{ULC}_{q,c}$ or $\mathrm{ULC}_{\mathrm{nc},c}$ than to prove either UGC or qUGC. Conversely, devising a polynomial-time approximation algorithm for $\mathrm{ULC}_{c,q}$ seems more achievable than disproving either UGC or qUGC. Note that $\mathrm{ULC}_{c,q}$ is in NP (since $\mathrm{ULC}_{c,c}$ is in NP).

1.3.4 Applications to Hardness of Approximation

The result by O'Donnell and Yuen [41] on the quantum value of 3XOR is stated in the next theorem.

▶ **Theorem 5.** *For every $\delta > 0$, the decision problem $3\mathrm{XOR}_q(1-\delta, \frac{1}{2}+\delta)$ is RE-hard.*

This result was originally proven via a reduction from $\mathrm{LC}_{\mathrm{nc}}$. However, a reduction from LC_q suffices with some straightforward modifications. The hardness of LC_q follows directly from the $\mathrm{MIP}^* = \mathrm{RE}$ theorem (as shown in Figure 4). Since a direct proof of LC_q is likely easier than the full proof of $\mathrm{MIP}^* = \mathrm{RE}$ [29], we propose Problem 5 in Section 1.6.

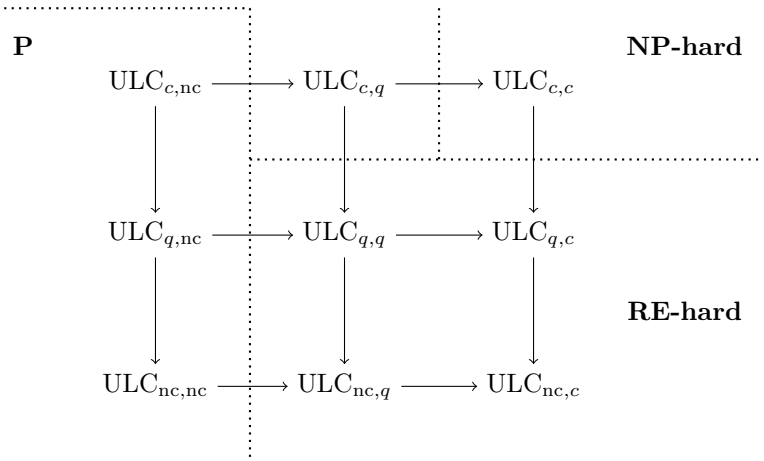
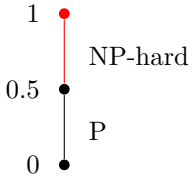
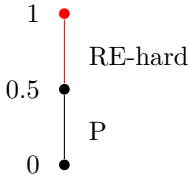


Figure 5 The complexity landscape of Unique-Label-Cover assuming both UGC and qUGC.



(a) Classical value [25].



(b) Quantum value [41].

Figure 6 Approximability of classical and quantum values of 3XOR: the vertical axis denotes approximation ratios between 0 and 1, with points of sharp transition in hardness (0.5 in both cases) indicated.

This theorem mirrors Håstad's celebrated result on the classical value of 3XOR [25]. Our current understanding of approximability of 3XOR is summarized in Figure 6.

In Sections 4 and 5 in the full version [38], we prove similar results for 2-Lin and MaxCut. Our reductions are from ULC$_q$ rather than LC$_q$. An informal overview of our MaxCut result is provided in the next section.

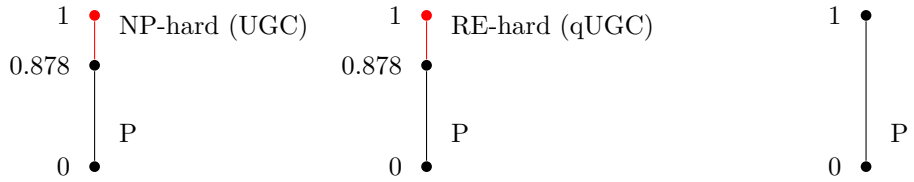
1.4 Result on MaxCut

How well can the classical, quantum, and noncommutative values of MaxCut be approximated? The current knowledge, including our result, is summarized in Figure 7.

The well-known Goemans-Williamson algorithm [21] for the classical value of MaxCut achieves an approximation ratio of $\alpha_{\mathrm{gw}} \approx 0.878$. More specifically, they showed that $\omega_c(G) \geq \alpha_{\mathrm{gw}}\omega_{\mathrm{sdp}}(G)$, where $\omega_{\mathrm{sdp}}(G)$ is the optimal value of a certain semidefinite programming (SDP) relaxation for MaxCut. Meanwhile, Tsirelson [50, 51] showed that $\omega_{\mathrm{nc}}(G) = \omega_{\mathrm{sdp}}(G)$. Combining these results yields the following chain of inequalities:

$$\alpha_{\mathrm{gw}}\omega_{\mathrm{sdp}}(G) \leq \omega_c(G) \leq \omega_q(G) \leq \omega_{\mathrm{nc}}(G) = \omega_{\mathrm{sdp}}(G). \tag{4}$$

From this chain of inequalities (see also Figure 8), we infer that the Goemans-Williamson algorithm is also a 0.878-approximation algorithm for the quantum value of MaxCut. But can we do better for the quantum value?



(a) Classical value [21, 34]. **(b)** Quantum value (this paper). **(c)** Noncommutative value [50, 51].

Figure 7 Approximability of MaxCut across different types of values.

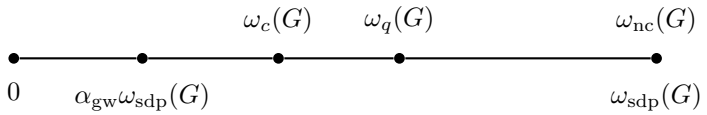


Figure 8 The MaxCut interval.

Under the assumption of the quantum Unique Games Conjecture introduced in this paper, we show:

The classical Goemans-Williamson algorithm is not only the best efficient approximation algorithm for the quantum value; it is the best approximation algorithm for the quantum value–regardless of efficiency.

▶ **Theorem 6** (Theorem 31 in the full version [38], informal)**.** *Assuming the quantum Unique Games Conjecture, it is* RE*-hard to approximate the quantum value of MaxCut to any ratio better than* α_{gw}*.*

Compare this with the following optimal inapproximability result from Khot et al. [34]:

▶ **Theorem 7** (KKMO, informal)**.** *Assuming the Unique Games Conjecture, it is* NP*-hard to approximate the classical value of MaxCut to any ratio better than* α_{gw}*.*

Integrality Gap. Similar to the Goemans-Williamson SDP relaxation, the quantum value–though undecidable–serves as a relaxation to the classical value. A natural question arises: how close is this quantum relaxation to the classical value? Can the quantum value outperform the SDP value in approximating the classical value?

Under the assumption of the quantum Unique Games Conjecture, we show that the SDP and quantum relaxations are incomparable. This is illustrated in Figure 9 and formally established in Theorem 35 in the full version [38]. Constructing examples that exhibit each of the extremes shown in Figure 9 remains an open problem.

1.5 Proof Ideas

In the conventional NP proof system for Label-Cover, the certificate is simply the list of labels for the vertices. This is what we called *classical assignment* in Section 1.2. The verifier's task in this proof system is to check that the assigned labels satisfy the edge constraints.

Håstad's gap-preserving reduction from Label-Cover (used, for instance, in proving the hardness of 3SAT and 3XOR) is best viewed as the construction of a more efficient version of this NP proof system, known as a probabilistically checkable proof (PCP). The PCP certificate for Label-Cover is an encoding of the classical assignment in binary using an error-correcting code, commonly referred to as the Long-Code, pioneered in [7]. In addition

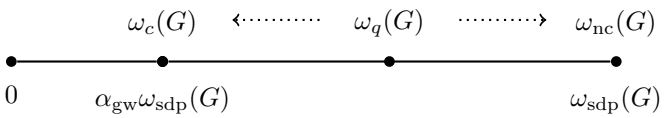


Figure 9 There exist instances realizing both extremes depicted in this picture. That is, among the integrality gap instances of MaxCut (see Feige and Schechtman [18]), i.e. those instances with $\omega_c(G) = \alpha_{\mathrm{gw}}\omega_{\mathrm{sdp}}(G)$, there are instances for which $\omega_q(G) = \omega_c(G)$ and instances for which $\omega_q(G) = \omega_{\mathrm{sdp}}(G)$.

to verifying that the labels in the encoding satisfy the edge constraints, the PCP verifier must also check that the proof consists of valid codewords. The soundness of this PCP relies on Fourier analysis. This was one of the key insights of Håstad's celebrated work.

In our case, we are reducing Unique-Label-Cover to 2-Lin (Section 4 in the full version [38]) and MaxCut (Section 5 in the full version [38]), but the core principles remain the same. Our reductions are based on the classical reductions by Khot [32] and KKMO [34], respectively. There are, however, several key differences, which we discuss in the rest of this section. For concreteness, we focus on the reduction from Unique-Label-Cover to MaxCut.

Completeness. This step requires a generalization of the Long-Code to encode quantum assignments. This is a straightforward generalization of the Long-Code to the operator setting and it is due to O'Donnell and Yuen [41]. The main challenge in proving completeness is ensuring that the encoded quantum assignment satisfies the commutation requirements of a quantum MaxCut assignment.

Soundness. The proof of soundness, as is customary, is done contrapositively. We begin with a quantum assignment for the MaxCut instance that achieves a large value, and from this, we construct a quantum assignment that has a sufficiently large value in the Unique-Label-Cover instance.

The classical proof constructs a probabilistic labeling based on the Fourier coefficients of the MaxCut assignment. Given a MaxCut assignment f, the labeling samples a subset of labels S with probability proportional to the squared Fourier coefficients, $\hat{f}(S)^2$. It then probabilistically selects a label $a \in S$ for each vertex. In the quantum setting, these squared Fourier weights are positive semidefinite operators. Defining $P^a := \sum_{S:a\in S} \frac{1}{|S|}\hat{f}(S)^2$ for each label a results in a *POVM measurement* for each vertex in the Unique-Label-Cover instance (see Section 2.2 in the full version [38] for the definition of POVMs). This is also due to [41].

These measurements satisfy the commutation relations required for a quantum assignment. However, they are not projective measurements, as required by definition of quantum assignment. Naimark's dilation theorem (see, for example, Watrous [52]) is typically used in such cases to convert POVMs into projective measurements. However, due to the commutation requirements (and the use of tracial states), Naimark's theorem is not applicable. Therefore, we need to projectivize these POVMs while preserving the commutation relations, which we accomplish using the projectivization lemma proved in Section 6 in the full version [38].

Classically, the proof of soundness relies on a deep result from Boolean Fourier analysis, known as the Majority-Is-Stablest [36], which relates Boolean functions to their Fourier weights. If the operators in our assignments were simultaneously diagonalizable, we could apply this theorem directly, as in the classical proof. However, the inherent noncommutativity of quantum assignments prevents us from doing so, forcing us to take a new approach.

To explain the problem and our resolution at a high level, recall that our goal is to construct a quantum labeling for an instance of Unique-Label-Cover. Let u be a vertex from the left-vertex set, and let N_u denote its neighbors in the right-vertex set. In the operator setting, while each neighbor's operator individually commutes with u's, the operators assigned to the neighbors may not commute with each other. This noncommutativity is a fundamental obstacle.

We first encountered this issue earlier, when attempting to view quantum assignments as extensions of probabilistic assignments. In a classical probabilistic assignment, each vertex u is assigned a single probability measure P_u over the set of labels. In contrast, a quantum assignment associates two probability measures, $P_{u,e}$ and $P_{v,e}$, with each edge $e = (u, v)$. The crux of the issue is that different edges incident on u – say, e and e' – can impose vastly different probability measures on u, which may conflict with each other.

This same phenomenon arises in the proof of soundness, except it manifests not in the probability measures, but in the Fourier coefficients of the quantum assignment. Classical soundness proofs organize around sets of neighbors, but this approach is too coarse to be applicable to the operator setting. The key to resolving this issue is to structure the proof around individual edges; this allows us to manage the noncommutative nature of the operator assignments without the sort of conflicts we mentioned earlier.

1.6 Future Directions

There are many open problems concerning the quantum value of CSPs and the conjectures we have proposed regarding Unique-Label-Cover:

1. **Quantizing Classical Reductions:** The works of O'Donnell and Yuen [41], as well as this paper, establish straightforward translations of some of the most well-known classical inapproximability results to the quantum setting. Could this indicate a more general phenomenon? More precisely, can any classical gap-preserving reduction from Label-Cover (or its variants like Unique-Label-Cover or Smooth-Label-Cover) be extended to a reduction for the quantum value?
2. **UGC versus qUGC:** What is the relationship between the classical UGC and the quantum UGC? If UGC is proven via a reduction from Label-Cover, can this reduction be "quantized" to prove qUGC? A positive answer to the previous question would also resolve this one affirmatively.
3. **Variants of qUGC:** In the classical setting, several formulations of UGC have been shown to be equivalent (see Section 3 in [33] and references therein). Could similar equivalences hold for the quantum variants discussed in this paper? See Section 3.2 in the full version [38] for further discussion of these qUGC variants.
4. **Noncommutative CSPs:** This work focuses primarily on the quantum value of CSPs. Could variants of qUGC help resolve hardness of approximation questions for the noncommutative value? The noncommutative value, which also plays an important role in quantum information, exhibits some intriguing behavior. As shown in Figure 7, unlike the classical or quantum value, the noncommutative value of MaxCut can be computed in polynomial time. However, it is known that the noncommutative value of Max-3-Cut is undecidable [28, 29]. [12] provides a 0.864-approximation algorithm for the noncommutative value of Max-3-Cut. We suspect that a variant of qUGC could prove useful in establishing the optimality of this algorithm.
5. **Weaker Version of MIP* = RE:** Is there a direct way to prove the hardness of approximating the quantum value of Label-Cover without relying on MIP* = RE? One potential approach is to quantize Dinur's proof of the classical PCP theorem [3, 2, 45, 13].

The commutation relations in LC_q could make this quantization feasible. Could this weaker result also have some of the implications of $\mathrm{MIP}^* = \mathrm{RE}$ such as the resolution of the Connes Embedding Problem?

6. **Integrality Gap Instances for MaxCut:** What is the smallest ratio between the classical and quantum values across all MaxCut instances? Assuming quantum UGC, we show this ratio is the Goemans-Williamson constant $\alpha_{\mathrm{gw}} \approx 0.878$ (Theorem 35 in the full version [38]). Can we construct an instance attaining this ratio?
7. **Label-Cover for QMA:** Approximating the classical value of Label-Cover is NP-complete (PCP theorem). Similarly, approximating the noncommutative and quantum values of Label-Cover is RE-complete ($\mathrm{MIP}^* = \mathrm{RE}$ theorem). What version of Label-Cover or Unique-Label-Cover naturally captures QMA? Could further constraints on the variations of Label-Cover and Unique-Label-Cover explored in this paper lead to suitable candidates? A positive resolution to this problem would answer the Quantum Games PCP Conjecture [39].
8. **Hardness of Approximation in Hamiltonian Complexity:** In quantum information theory, quantum MaxCut often refers to the anti-ferromagnetic Heisenberg model, a special case of the local Hamiltonian problem. Although [26] attempts to prove UGC hardness for the anti-ferromagnetic Heisenberg model, reductions from UGC only establish NP-hardness. Could variations of qUGC yield stronger results?

   While the inapproximability of the anti-ferromagnetic Heisenberg model is one of our motivations for proposing the quantum UGC, our current methods do not yet apply to the inapproximability of local Hamiltonian problems.
9. **Gadget Reductions:** In classical complexity theory, gadgets [25, 49] are widely used to demonstrate hardness of approximation. Can this method be "quantized"? [22, 28] are examples of gadget reductions in the quantum setting.

Organization of the Full Version of the Paper [38]

In Section 2, we present notations, definitions, and some Fourier analytic results that underpin this work. Section 2.1 specifically addresses the notations used throughout the paper. In Section 2.2, we introduce the concept of quantum measurement and discuss several definitions related to projective measurements. Section 2.3 reviews concepts such as reductions and decision problems, and Section 2.4 summarizes results from Fourier analysis.

Section 3.1 offers the most general definition of CSPs necessary for this work. It also covers the classical, quantum, and noncommutative values of a CSP instance, providing definitions which are crucial for the remainder of the paper. In Section 3.2, we discuss variants of quantum UGC. Section 4 contains the proof of our theorem on the quantum value of 2-Lin, and Section 5 presents the proof of our theorem on the quantum value of MaxCut. Finally, two technical lemmas are given in Sections 6 and 7.

References

1 Sanjeev Arora, László Babai, Jacques Stern, and Z Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54(2):317–331, 1997. doi:10.1006/jcss.1997.1472.

2 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, May 1998. doi:10.1145/278298.278306.

3 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, January 1998. doi:10.1145/273865.273901.

4 László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*, STOC '91, pages 21–32, New York, NY, USA, 1991. Association for Computing Machinery. doi:10.1145/103418.103428.

5 László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1(1):3–40, March 1991. doi:10.1007/BF01200056.

6 John Stewart Bell. On the Einstein Podolsky Rosen paradox. *Physics*, 1(3):195–200, 1964. doi:10.1103/PhysicsPhysiqueFizika.1.195.

7 Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs and non-approximability-towards tight results. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 422–431, 1995. doi:10.1109/SFCS.1995.492573.

8 Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 113–131, New York, NY, USA, 1988. ACM. doi:10.1145/62212.62223.

9 John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23:880–884, October 1969. doi:10.1103/PhysRevLett.23.880.

10 Richard Cleve, Peter Hoyer, Benjamin Toner, and John Watrous. Consequences and limits of nonlocal strategies. In *Proceedings of the 19th IEEE Annual Conference on Computational Complexity*, CCC '04, pages 236–249, Washington, DC, USA, 2004. IEEE Computer Society. doi:10.1109/CCC.2004.9.

11 Richard Cleve and Rajat Mittal. Characterization of binary constraint system games. In *International Colloquium on Automata, Languages, and Programming (ICALP) 2012*, pages 320–331, 2012. doi:10.1007/978-3-662-43948-7_27.

12 Eric Culf, Hamoon Mousavi, and Taro Spirig. Approximation algorithms for noncommutative constraint satisfaction problems, 2023. doi:10.48550/arXiv.2312.16765.

13 Irit Dinur. The pcp theorem by gap amplification. *J. ACM*, 54(3):12–es, June 2007. doi:10.1145/1236457.1236459.

14 Irit Dinur, David Steurer, and Thomas Vidick. A parallel repetition theorem for entangled projection games. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 197–208, 2014. doi:10.1109/CCC.2014.28.

15 Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *J. ACM*, 43(2):268–292, March 1996. doi:10.1145/226643.226652.

16 Uriel Feige and Joe Kilian. Two-prover protocols—low error at affordable rates. *SIAM Journal on Computing*, 30(1):324–346, 2000. doi:10.1137/S0097539797325375.

17 Uriel Feige and László Lovász. Two-prover one-round proof systems: their power and their problems (extended abstract). In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '92, pages 733–744, New York, NY, USA, 1992. Association for Computing Machinery. doi:10.1145/129712.129783.

18 Uriel Feige and Gideon Schechtman. On the optimality of the random hyperplane rounding technique for max cut. *Random Structures & Algorithms*, 20(3):403–440, 2002. doi:10.1002/rsa.10036.

19 Lance Fortnow, John Rompel, and Michael Sipser. On the power of multi-prover interactive protocols. *Theoretical Computer Science*, 134(2):545–557, 1994. doi:10.1016/0304-3975(94)90251-8.

20 Tobias Fritz. Tsirelson's problem and Kirchberg's conjecture. *Reviews in Mathematical Physics*, 24(05):1250012, 2012. doi:10.1142/S0129055X12500122.

21 Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, November 1995. doi:10.1145/227683.227684.

22 Samuel J. Harris. Universality of Graph Homomorphism Games and the Quantum Coloring Problem. *Annales Henri Poincare*, 25(10):4321–4356, 2024. doi:10.1007/s00023-024-01422-5.

23 John William Helton, Kyle P Meyer, Vern I Paulsen, and Matthew Satriano. Algebras, synchronous games and chromatic numbers of graphs. *arXiv preprint*, 2017. doi:10.48550/arXiv.1703.00960.

24 Johan Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 627–636, 1996. doi:10.1109/SFCS.1996.548522.

25 Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, July 2001. doi:10.1145/502090.502098.

26 Yeongwoo Hwang, Joe Neeman, Ojas Parekh, Kevin Thompson, and John Wright. *Unique Games hardness of Quantum Max-Cut, and a conjectured vector-valued Borell's inequality*, pages 1319–1384. Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2023. doi:10.1137/1.9781611977554.ch48.

27 Tsuyoshi Ito, Hirotada Kobayashi, and Keiji Matsumoto. Oracularization and two-prover one-round interactive proofs against nonlocal strategies. *2009 24th Annual IEEE Conference on Computational Complexity*, pages 217–228, 2008. doi:10.1109/CCC.2009.22.

28 Zhengfeng Ji. Binary constraint system games and locally commutative reductions, 2013. doi:10.48550/arXiv.1310.3794.

29 Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. $\mathsf{MIP}^* = \mathsf{RE}$. *Commun. ACM*, 64(11):131–138, October 2021. doi:10.1145/3485628.

30 Marius Junge, Miguel Navascues, Carlos Palazuelos, David Perez-Garcia, Volkher B. Scholz, and Reinhard F. Werner. Connes' embedding problem and tsirelson's problem. *Journal of Mathematical Physics*, 52(1):012102, 2011. doi:10.1063/1.3514538.

31 Julia Kempe, Oded Regev, and Ben Toner. Unique games with entangled provers are easy. *SIAM Journal on Computing*, 39(7):3207–3229, 2010. doi:10.1137/090772885.

32 Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings 17th IEEE Annual Conference on Computational Complexity*, pages 25–, 2002. doi:10.1109/CCC.2002.1004334.

33 Subhash Khot. On the unique games conjecture. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 3–, 2005. doi:10.1109/SFCS.2005.61.

34 Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007. doi:10.1137/S0097539705447372.

35 Se-Jin Kim, Vern Paulsen, and Christopher Schafhauser. A synchronous game for binary constraint systems. *Journal of Mathematical Physics*, 59(3):032201, 2018. doi:10.1063/1.4996867.

36 Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences: Invariance and optimality. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 21–30, 2005. doi:10.1109/SFCS.2005.53.

37 Hamoon Mousavi, Seyed Sajjad Nezhadi, and Henry Yuen. Nonlocal games, compression theorems, and the arithmetical hierarchy. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, pages 1–11, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3519935.3519949.

38 Hamoon Mousavi and Taro Spirig. A quantum unique games conjecture, 2024. doi:10.48550/arXiv.2409.20028.

39 Anand Natarajan and Chinmay Nirkhe. The status of the quantum pcp conjecture (games version), 2024. doi:10.48550/arXiv.2403.13084.

40 Anand Natarajan and John Wright. NEEXP $\subseteq$ MIP*. In *IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 510–518, 2019. doi:10.48550/arXiv.1904.05870.

41 Ryan O'Donnell and Henry Yuen. Private communication, 2024.

42 Narutaka Ozawa. About the Connes embedding conjecture: Algebraic approaches. *Jpn. J. Math.*, 8:147–183, 2013. doi:10.1007/s11537-013-1280-5.

43 Vern I Paulsen, Simone Severini, Daniel Stahlke, Ivan G Todorov, and Andreas Winter. Estimating quantum chromatic numbers. *Journal of Functional Analysis*, 270(6):2188–2222, 2016. doi:10.1016/j.jfa.2016.01.010.

44 Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 245–254, New York, NY, USA, 2008. Association for Computing Machinery. doi:10.1145/1374376.1374414.

45 Ran Raz. A parallel repetition theorem. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '95, pages 447–456, New York, NY, USA, 1995. Association for Computing Machinery. doi:10.1145/225058.225181.

46 Volkher B Scholz and Reinhard F Werner. Tsirelson's problem, 2008. doi:10.48550/arXiv.0812.4305.

47 William Slofstra. The set of quantum correlations is not closed. In *Forum of Mathematics, Pi*, volume 7. Cambridge University Press, 2019. doi:10.1017/fmp.2018.3.

48 William Slofstra. Tsirelson's problem and an embedding theorem for groups arising from non-local games. *Journal of the American Mathematical Society*, 2019. doi:10.1090/jams/929.

49 Luca Trevisan, Gregory B. Sorkin, Madhu Sudan, and David P. Williamson. Gadgets, approximation, and linear programming. *SIAM Journal on Computing*, 29(6):2074–2097, 2000. doi:10.1137/S0097539797328847.

50 Boris Tsirelson. Quantum analogues of the Bell inequalities. the case of two spatially separated domains. *Journal of Soviet Mathematics*, 36(4):557–570, 1987. doi:doi:10.1007/BF01663472.

51 Boris Tsirelson. Some results and problems on quantum Bell-type inequalities. *Hadronis Journal Supplement*, 8:320–331, 1993. URL: https://www.tau.ac.il/~tsirel/download/hadron.pdf.

52 John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018. doi:10.1017/9781316848142.

Sketching, Moment Estimation, and the Lévy-Khintchine Representation Theorem

Seth Pettie
University of Michigan, Ann Arbor, MI, USA

Dingyu Wang
University of Michigan, Ann Arbor, MI, USA

Abstract

In the d-dimensional turnstile streaming model, a frequency vector $\mathbf{x} = (\mathbf{x}(1), \ldots, \mathbf{x}(n)) \in (\mathbb{R}^d)^n$ is updated entry-wisely over a stream. We consider the problem of *f-moment estimation* for which one wants to estimate

$$f(\mathbf{x}) \stackrel{\text{def}}{=} \sum_{v \in [n]} f(\mathbf{x}(v))$$

with a small-space sketch. A function f is *tractable* if the f-moment can be estimated to within a constant factor using $\text{polylog}(n)$ space.

The f-moment estimation problem has been intensively studied in the $d = 1$ case. Flajolet and Martin estimate the F_0-moment ($f(x) = \mathbb{1}(x > 0)$, incremental stream); Alon, Matias, and Szegedy estimate the L_2-moment ($f(x) = x^2$); Indyk estimates the L_α-moment ($f(x) = |x|^\alpha$), $\alpha \in (0, 2]$. For $d \geq 2$, Ganguly, Bansal, and Dube estimate the $L_{p,q}$ hybrid moment ($f : \mathbb{R}^d \to \mathbb{R}, f(x) = (\sum_{j=1}^d |x_j|^p)^q$), $p \in (0, 2], q \in (0, 1)$. For tractability, Bar-Yossef, Jayram, Kumar, and Sivakumar show that $f(x) = |x|^\alpha$ is *not* tractable for $\alpha > 2$. Braverman, Chestnut, Woodruff, and Yang characterize the class of tractable one-variable functions except for a class of *nearly periodic functions*.

In this work we present a simple and generic scheme to construct sketches with the novel idea of hashing indices to *Lévy processes*, from which one can estimate the f-moment $f(\mathbf{x})$ where f is the *characteristic exponent* of the Lévy process. The fundamental *Lévy-Khintchine representation theorem* completely characterizes the space of all possible characteristic exponents, which in turn characterizes the set of f-moments that can be estimated by this generic scheme.

The new scheme has strong explanatory power. It unifies the construction of many existing sketches (F_0, L_0, L_2, L_α, $L_{p,q}$, etc.) and it implies the tractability of many nearly periodic functions that were previously unclassified. Furthermore, the scheme can be conveniently generalized to multidimensional cases ($d \geq 2$) by considering multidimensional Lévy processes and can be further generalized to estimate *heterogeneous moments* by projecting different indices with different Lévy processes. We conjecture that the set of tractable functions can be characterized using the Lévy-Khintchine representation theorem via what we called the *Fourier-Hahn-Lévy* method.

2012 ACM Subject Classification Theory of computation → Sketching and sampling

Keywords and phrases Streaming Sketches, Lévy Processes

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.77

Related Version *Full Version*: https://arxiv.org/abs/2410.17426

Funding This work was supported by NSF Grant CCF-2221980.

1 Introduction

Streaming algorithms have been studied for over 40 years, beginning with Morris's [25] *approximate counter* and Munro and Patterson's [26] 1- and 2-pass *selection* algorithms, both from 1978. In 1983 Flajolet and Martin [14] designed and analyzed the first "modern"


© Seth Pettie and Dingyu Wang;
licensed under Creative Commons License CC-BY 4.0



sketch called *Probabilistic Counting with Stochastic Averaging* (PCSA). PCSA works only in *incremental* streams whereas all linear sketches, such as the celebrated AMS sketch [1] for estimating L_2-moments, maintain a vector of elements subject to both increments and decrements. In this work we consider a generalized turnstile model, where each element has a value in $\mathbb{R}^d$ for some dimension $d \in \mathbb{Z}_+$.

▶ **Definition 1** ($\mathbb{R}^d$-turnstile model)**.** *Let $[n] = \{1, 2, \ldots, n\}$ be the universe. The frequency vector $\mathbf{x} = (\mathbf{x}(1), \ldots, \mathbf{x}(n)) \in (\mathbb{R}^d)^n$ is initialized as all zeros and gets updated by a stream of pairs of the form of (v, y), where $v \in [n]$ and $y \in \mathbb{R}^d$.*

- Update(v, y), $\mathbf{x}(v) \leftarrow \mathbf{x}(v) + y$.

In this paper we consider the following generic streaming estimation problem.

▶ **Problem 1** (f-moment estimation in the $\mathbb{R}^d$-turnstile model)**.** Fix $d \in \mathbb{Z}_+$. Let $f : \mathbb{R}^d \to \mathbb{C}$ with $f(0) = 0$. Let $\mathbf{x} \in (\mathbb{R}^d)^n$ be the frequency vector of the current stream. Estimate the *f-moment* $f(\mathbf{x})$, where

$$f(\mathbf{x}) \stackrel{\text{def}}{=} \sum_{v \in [n]} f(\mathbf{x}(v)).$$

The f-moment estimation problem was originally posed by Alon, Matias, and Szegedy [1].[1] They estimated the L_2-moment ($f(x) = |x|^2$) with a data structure now commonly known as the AMS sketch, and then asked in general what f-moments can be estimated in polylog space. Indyk [17] designed a class of sketches for estimating the L_p moments, with $p \in (0, 2]$. When $p > 2$, Bar-Yossef, Jayram, Kumar, and Sivakumar [2] proved that estimating the L_p-moment requires $\Omega(\text{poly}(n))$ space. The L_0-moment ($f(x) = \mathbb{1}(x \neq 0)$) has been estimated in two distinct ways. Cormode, Data, Indyk, and Muthukrishnan [10] approximate L_0 by L_α with very small $\alpha > 0$. Kane, Nelson, and Woodruff [20] project each element onto $\mathbb{Z}_p$, $p > \epsilon^{-1} \log n$ being a random prime, which effectively reduces L_0-estimation to cardinality estimation over incremental streams (F_0-estimation).

Apart from the frequency moments L_p, $p \in [0, 2]$, there are other function moments that can be estimated over streams. In particular, a function f is called *tractable* if the f-moment can be approximated to within a $1 \pm \epsilon$ factor in $\text{poly}(\epsilon^{-1}, \log n)$ space. Characterizing the class of tractable functions (the *tractability problem*) has been a central theoretical problem in the streaming literature. Braverman and Ostrovsky [6] managed to characterize all tractable functions $f : \mathbb{Z} \to \mathbb{R}_+$ that are symmetric ($f(x) = f(-x)$) and increasing on $[0, \infty)$. Braverman and Chestnut [3] then charaterized all tractable functions that are symmetric, non-negative, and *decreasing* on $[1, \infty)$. Braverman, Chestnut, Woodruff, and Yang [4] extended the characterization to all symmetric functions, except for a class of "nearly periodic" functions.

The tractability question is closely related to the problem of designing *universal sketches*. A universal sketch specifies a family of functions $\mathcal{F}$ such that for *any* $f \in \mathcal{F}$, the f-moment can be estimated. Sketches in [6, 7, 4] are all universal for their respective classes; they are based on sampling L_2-heavy hitters from subsampled versions of the input vector, then applying f to the samples. Though in this work we focus on the polylog-space regime, there have also been intensive studies for sketching problems in the polynomial-space regime as well, e.g., L_p-moment with $p > 2$ [1, 5, 18, 8, 28, 15], cascaded norms [19], and Schatten norms [24].

[1] Unless specified otherwise, we assume the stream is one-dimensional and integral, i.e., $\mathbf{x} \in \mathbb{Z}^n$, when discussing prior works.

While most of the literature considers the one-dimensional case where $d = 1$, Ganguly, Bansal, and Dube [16] estimate the $L_{p,q}$ hybrid moment, which is defined to be $f(x_1, \ldots, x_d) = (\sum_{j=1}^{d} |x_j|^p)^q$, where $p \in [0, 2], q \in [0, 1]$. The $L_{p,q}$-moment is a meaningful class of multidimensional moments which include, e.g., the Euclidean length $f(x_1, \ldots, x_d) = \sqrt{\sum_{j=1}^{d} x_j^2}$, corresponding to the $L_{2,1/2}$-moment. If the elements are points in $\mathbb{R}^d$ subject to a stream of updates of the form of `(key, displacement)`, then the $L_{2,1/2}$-moment is equal to the sum of the Euclidean distances $\sum_{v\in[n]} |\mathbf{x}(u)|$. Multidimensional inputs can arise when querying *multiple* one-dimensional streams. For example, suppose we have two $\mathbb{R}$-turnstile streams with frequency vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. A query like "how many elements are non-zero in $\mathbf{x}$ and at least 3 in $\mathbf{y}$" can be written as a two-dimensional f-moment query with $f(x, y) = \mathbb{1}(x \neq 0)\,\mathbb{1}(y \geq 3)$.

Main Contribution

In this work, we uncover an intimate relation between turnstile sketches and *Lévy processes*. Lévy processes have been studied since the early 20th century, and have been used to model phenomena in various fields, e.g., physics (how does a gas particle move?) and finance (how does the stock price change?). We will show that all Lévy processes have *algorithmic* interpretations in the context of streaming sketches, and that the fundamental *Lévy-Khintchine representation theorem* leads to a unified view of sketching for moment estimation.

We give a detailed technical synopsis of Lévy processes in Section 2. For the time being, a Lévy process $(X_t)_{t\geq 0}$, where $X_t \in \mathbb{R}^d$, is defined by having independent, stationary increments. I.e., $(X_{t_1+t_2} - X_{t_1}) \sim X_{t_2}$, for any $t_1, t_2 \in \mathbb{R}_+$, and $X_{t_1}, X_{t_2} - X_{t_1}, \ldots, X_{t_k} - X_{t_{k-1}}$ are mutually independent for any $t_1 < \cdots < t_k$. Some common one-dimensional Lévy processes are

Linear drift. $X_t = \gamma t$ with drift rate γ.
Brownian motion. $X_t \sim \mathcal{N}(0, t\sigma^2)$ with variance σ^2.
Poisson process. $X_t \sim \text{Poisson}(\lambda t)$ with rate λ.
α-stable process. Defined by $X_t/t^{1/\alpha} \sim X_1 \sim \alpha$-stable. (For $\alpha \in (0, 2]$, the α-stable random variable X_1 on $\mathbb{R}$ is defined by its characteristic function $\mathbb{E}e^{izX_1} = e^{-|z|^\alpha}$, for any $z \in \mathbb{R}$.)

These can all be generalized to higher dimensions. The Lévy-Khintchine representation theorem identifies every Lévy process X with its *characteristic exponent* $f_X : \mathbb{R}^d \to \mathbb{C}$, where for any $z \in \mathbb{R}^d$, $\mathbb{E}\exp(i\langle X_t, z\rangle) = \exp(-tf_X(z))$.

1.1 Lévy Processes and Sketching

We briefly explain why Lévy processes naturally lie at the heart of linear sketching. Suppose X is the random memory state of a *linear* sketch[2] of input vector $\mathbf{x}$ and $\hat{f}(X)$ is an estimate of the f-moment $f(\mathbf{x})$. Now consider the situation where the input is repeated w times over disjoint domains. Then by construction, the f-moment becomes $w \cdot f(\mathbf{x})$ and the estimate becomes $\hat{f}(X^{(1)} + \cdots + X^{(w)})$ where $X^{(j)}$ are i.i.d. copies of X.[3] In other words, we should have

$$\frac{\hat{f}(X^{(1)} + \cdots + X^{(w)})}{w} \approx f(\mathbf{x}),$$

[2] It suffices to consider linear sketches in the turnstile model [23].
[3] One can think of this as increasing the vector length from n to wn and letting $X^{(j)}$ be the sketch of $\mathbf{x}^{(j)}$, which is a copy of $\mathbf{x}$ on the index interval $[(j-1)n, jn)$.

for *any* $w \in \mathbb{Z}_+$ and also for the limiting case as $w \to \infty$. Therefore, no matter how complicated the distribution of the linear sketch may be, the sum $X^{(1)} + \cdots + X^{(w)}$ will converge[4] to some well-behaved limiting random process. For the AMS sketch [1], X happens to be sub-gaussian and therefore the normalized sum converges to a Gaussian. For Indyk's [17] stable sketch, X is α-stable, and therefore the normalized sum remains α-stable. These are merely two special cases of Lévy processes. In general, if $X = (X_t)_{t\in\mathbb{R}_+}$ is a Lévy process, then $X^{(1)} + \cdots + X^{(w)} = (X_t^{(1)} + \cdots + X_t^{(w)})_{t\in\mathbb{R}_+} \sim (X_{wt})_{t\in\mathbb{R}_+}$. Thus summing i.i.d. Lévy processes is equivalent to simply *rescaling time*. After normalizing the time scale,[5] we see Lévy processes are stable under i.i.d. sums. Lévy processes therefore form a mathematical closure of linear sketches in terms of their limiting distributions.

In practice it usually suffices to construct some algorithmically simple random projection in the *domain of attraction* of the limiting process. For example, the AMS sketch for estimating L_2 does not need to explicitly use Gaussians in its projection; Rademacher ($\{-1, 1\}$) random variables suffice. For a *prototypical solution*, it is convenient to consider linear sketches with projection sampled directly from Lévy processes. The distribution of the sketch will always lie in the space of Lévy processes and is therefore easier to track.

1.2 Random Projection with Lévy Processes

We now consider linear sketching with Lévy processes and show how different processes can be used to track different function moments. Throughout the paper we work in the *random oracle* model, in which we can evaluate uniformly random hash functions $H : [n] \to [0, 1]$.

Given any Lévy process $X = (X_t)_{t\in\mathbb{R}_+}$ on $\mathbb{R}^d$, by the Lévy-Khintchine representation theorem (Theorem 7), there exists a function $f_X : \mathbb{R}^d \to \mathbb{C}$ such that for any $t \in \mathbb{R}_+$ and $z \in \mathbb{R}^d$,

$$\mathbb{E}e^{i\langle z, X_t\rangle} = e^{-tf_X(z)}. \qquad \text{(Lévy-Khintchine)}$$

Suppose now we aggregate all the input in one register in the following way

$$C_t = \sum_{v\in[n]} \left\langle \mathbf{x}(v), X_t^{(v)} \right\rangle,$$

where the $X_t^{(v)}$ are i.i.d. copies of X_t. Clearly C_t is a linear sketch which can be maintained over a distributed stream. We thus have

$$\begin{aligned}
\mathbb{E}e^{iC_t} &= \mathbb{E}e^{i\sum_{v\in[n]}\left\langle X_t^{(v)}, \mathbf{x}(v)\right\rangle} \\
&= \prod_{v\in[n]} \mathbb{E}e^{i\left\langle X_t^{(v)}, \mathbf{x}(v)\right\rangle} && \text{(by independence)} \\
&= \prod_{v\in[n]} e^{-tf_X(\mathbf{x}(v))} && \text{(by Lévy-Khintchine)} \\
&= e^{-tf_X(\mathbf{x})}, && \text{(recall that } f(\mathbf{x}) = \textstyle\sum_{v\in[n]} f(\mathbf{x}(v))\text{)}
\end{aligned}$$

from which the f_X-moment $f_X(\mathbf{x})$ can be recovered by choosing a suitable time $t \approx \Theta(1/|f(\mathbf{x})|)$. Of course, we do not always know a suitable t, and therefore maintain samples for many t, evenly spaced on a logarithmic scale. This is the key observation that lets us

[4] with a proper normalization depending on X and w

[5] To see how this "time normalization" generalizes the typical scalar normalization of stable variables, note that for α-stable variables, the normalization is $w^{-1/\alpha}(X^{(1)} + \cdots + X^{(w)})$, which is equivalent to scale the time down by w since $w^{-1/\alpha}X_{wt} \sim X_t$ if $(X_t)_{t\in\mathbb{R}_+}$ is α-stable.

estimate every f-moment, where f is the characteristic exponent of a Lévy process. In addition, note that the statistic e^{iC_t} only depends on the value of $C_t \mod 2\pi$. Therefore, it suffices to store $C_t \mod 2\pi$ for the estimation scheme above, which is a *compact value* in the range $[0, 2\pi)$. See Figure 1.[6]

1.3 Stable Projections

A d-dimensional Lévy process X is called α-*stable* if for any $t \in \mathbb{R}_+$, $X_t \sim t^{1/\alpha} X_1$. If the Lévy process X in $C_t = \sum_{v\in[n]} \left\langle \mathbf{x}(v), X_t^{(v)} \right\rangle$ is α-stable, then we have

$$C_t = \sum_{v\in[n]} \left\langle \mathbf{x}(v), X_t^{(v)} \right\rangle \sim \sum_{v\in[n]} \left\langle \mathbf{x}(v), t^{1/\alpha} X_1^{(v)} \right\rangle \sim t^{1/\alpha} C_1.$$

In other words, the projection C_t is also α-stable. Due to this self-similarity, it suffices to consider the projection only at time 1 and there is no need for using multiple time scales. Now let the characteristic exponent be $f_X(z)$. Then we have the characteristic function

$$\mathbb{E} e^{izC_1} = \mathbb{E} e^{iz \sum_{v\in[n]} \left\langle \mathbf{x}(v), X_1^{(v)} \right\rangle} = \prod_{v\in[n]} \mathbb{E} e^{i\left\langle \mathbf{x}(v), zX_1^{(v)} \right\rangle} \qquad \text{(by independence)}$$

Since $X^{(v)}$ is α-stable, we have $zX_1^{(v)} \sim X_{z^\alpha}^{(v)}$

$$\begin{aligned}
&= \prod_{v\in[n]} \mathbb{E} e^{i\left\langle \mathbf{x}(v), X_{z^\alpha}^{(v)} \right\rangle} \\
&= \prod_{v\in[n]} e^{-z^\alpha f(\mathbf{x}(v))} && \text{(by Lévy-Khintchine)} \\
&= e^{-z^\alpha f(\mathbf{x})} = e^{-(z f(\mathbf{x})^{1/\alpha})^\alpha} \\
&= \mathbb{E} e^{iz f(\mathbf{x})^{1/\alpha} Y_\alpha}, && \text{(by Lévy-Khintchine)}
\end{aligned}$$

where Y_α is a *one*-dimensional α-stable random variable with $\mathbb{E} e^{izY_\alpha} = e^{-|z|^\alpha}$. Since the characteristic function uniquely identifies the distribution, we know $C_1 \sim f(\mathbf{x})^{1/\alpha} Y_\alpha$. Note that a register in Indyk's [17] L_α-stable sketch distributes as $(\sum_{v\in[n]} |\mathbf{x}(v)|^\alpha)^{1/\alpha} Y_\alpha$. Thus m i.i.d. copies of C_1 *emulate* the distribution of Indyk's L_α-sketch where the L_α-moment is replaced by the f-moment. Such function moments can be estimated by Indyk's estimator [17] or Li's estimators [22] with the same error guarantee.

Note that such stable projection takes less space than the generic projection considered in Section 1.2 since in general C_t needs to be stored for for many t while stable projection only requires to store C_1. Next, we formally state the new results.

1.4 New Results

The main theorem connects the Lévy-Khintchine representation theorem for generic Lévy processes with f-moment estimation in the $\mathbb{R}^d$-turnstile model. See Table 1 for some notation. Note that for a vector $\mathbf{x} \in (\mathbb{R}^d)^n$ with multidimensional entries ($d \geq 2$), the L_1 and L_∞ norms use the Euclidean norm per entry. For example, $\|\mathbf{x}\|_1 = \sum_{v\in[n]} |\mathbf{x}(v)|$, where $|\mathbf{x}(v)| = \sqrt{\sum_{j=1}^d \mathbf{x}(v)_j^2}$.

[6] Strictly speaking, by wrapping C_t in $[0, 2\pi)$, the sketch becomes non-linear. It is still additive, and therefore allows simple merging of two sketches, but does not permit scalar multiplication.

Table 1 Notation.

Notation	Definition	Notes
$\lvert x\rvert$	$(\sum_{j=1}^{d} x_j^2)^{1/2}$	$x \in \mathbb{R}^d$, Euclidean norm
$\lvert z\rvert$	$(z\bar{z})^{1/2}$	$z \in \mathbb{C}$, modulus
$\lVert\mathbf{x}\rVert_0$	$\sum_{v\in[n]} \mathbb{1}\left(\mathbf{x}(v) \neq 0\right)$	$\mathbf{x} \in (\mathbb{R}^d)^n$, L_0-moment
$\lVert\mathbf{x}\rVert_1$	$\sum_{v\in[n]} \lvert\mathbf{x}(v)\rvert$	$\mathbf{x} \in (\mathbb{R}^d)^n$, L_1-moment
$\mathbb{T}$	complex unit circle	identified by $[0, 2\pi)$
$\mathbb{S}_{d-1}$	$\{x \in \mathbb{R}^d : \lvert x\rvert = 1\}$	unit sphere in $\mathbb{R}^d$
$\mathbb{V}Z$	$\mathbb{E}(Z - \mathbb{E}Z)\overline{(Z - \mathbb{E}Z)}$	variance of Z

▶ **Theorem 2** (generic Lévy-Tower, Section 3). *Let $f : \mathbb{R}^d \to \mathbb{C}$ be any function of the form*

$$f(z) = \frac{1}{2}\langle z, Az\rangle - i\langle\gamma, z\rangle + \int_{\mathbb{R}^d} \left(1 + i\langle z, s\rangle \mathbb{1}\left(|s| < 1\right) - e^{i\langle z,s\rangle}\right) \nu(ds),$$

where A is a covariance matrix, $\gamma \in \mathbb{R}^d$, and ν is a positive measure over $\mathbb{R}^d$ with $\int_{\mathbb{R}^d} \min(|x|^2, 1)\, \nu(dx) < \infty$. There exists a mergeable sketch of $O(\epsilon^{-2} \log n)$ words such that for any input stream $\mathbf{x} \in (\mathbb{R}^d)^n$ with $|f(\mathbf{x})| \in [1, \mathrm{poly}(n)]$, it returns an estimate $\widehat{f(\mathbf{x})}$ that with probability 99/100, where

$$\left|\widehat{f(\mathbf{x})} - f(\mathbf{x})\right| \leq O(\epsilon |f(\mathbf{x})|).$$

▶ Remark 3. The Lévy-Tower sketch improves the understanding of tractability of one-dimension function moments. In particular, it implies the tractability of many nearly periodic functions that were not previously classified. See Section 5 for a discussion of the new result and the existing tractability results. It also implies the tractability of a large class of *multidimensional moments* which have not been considered before.

As discussed in Section 1.3, stable projections form a special subclass of this linear projection method. We prove the following *emulation theorem*, enabling us to extend any analysis of Indyk's [17] classic L_α-stable sketch to the estimation of more general stable moments.

▶ **Theorem 4** (Lévy-Stable, Section 3.3). *Let f be the characteristic exponent of any d-dimensional α-stable process X. When $\alpha = 2$, $f(x) = e^{-\frac{1}{2}t\langle x, Ax\rangle}$, where A is a covariance matrix, and when $\alpha \in (0, 2)$, $f(x) = \int_{\mathbb{S}_{d-1}} |\langle x, \xi\rangle|^\alpha \, \mu(d\xi)$, where μ is a symmetric, positive measure on $\mathbb{S}_{d-1}$.*[7] *Let the Lévy-Stable sketch be parameterized by X. Given any input vector $\mathbf{x} \in (\mathbb{R}^d)^n$ and $\mathbf{x}' \in \mathbb{R}^n$ such that $f(\mathbf{x}) = \sum_{v\in[n]} |\mathbf{x}'(v)|^\alpha$, Lévy-Stable with input $\mathbf{x}$ and Indyk's L_α-stable sketch with input $\mathbf{x}'$ distribute identically.*

Theorem 4 serves as an illuminating example showing how the Lévy-Khintchine theorem helps to understand streaming sketching. Previously, only two classes of stable moments are considered: one dimensional L_α-moments by Indyk [17], and multidimensional $L_{p,q}$-moments by Ganguly et al. [16], which are sketched by *algorithmic tricks* of combining stable random variables. The Lévy-Stable sketch extends such tricks to *all stable processes* in a systematic way. For example, we can now estimate, using any estimator of Indyk's sketch [17, 22], the f-moment of an $\mathbb{R}^3$-turnstile stream with

[7] For the case $\alpha \in (0, 2)$, only symmetric processes are considered here for simplicity.

$$f(x) = \int_{\mathbb{S}_2} \frac{|\langle \xi, x \rangle|}{|\xi_1|^2 + |\xi_2| + |\xi_3|^{1/2}} \, d\xi,$$

where $\mathbb{S}_2$ is the unit $\mathbb{R}^3$-sphere.

Table 2 Sketches and Lévy processes. The target function f_X is the characteristic exponent of the corresponding Lévy process, i.e., $f_X(x) = -\log \mathbb{E} e^{i\langle x, X_1 \rangle}$. Random oracle is assumed (i.e., hash functions are provided externally) and the space is measured in *words* where each word uses $O(\log n)$ bits. $O(\log n)$-size words suffice for the Lévy-Tower and Lévy-Stable sketches under the common assumption that the total number of updates is at most $\text{poly}(n)$. Since F_0 is the cardinality over incremental stream, the space can be compressed to $O(\epsilon^{-2})$ *bits* [20, 21, 27]. In [20], L_0 is approximated by $\sum_{v \in [n]} \mathbb{1}(P | \mathbf{x}(v))$ where P is a random prime. In [10], L_0 is approximated by $\sum_{v \in [n]} |\mathbf{x}(v)|^\alpha$ with a small α. See [17, 20] for space complexity for estimating F_0, L_0, L_α when hash functions are stored explicitly.

Sketch Citation	Target $f(x)$	Space	Corresponding Lévy Process
Flajolet & Martin [14] (F_0)	$\mathbb{1}(x > 0)$	$O(\epsilon^{-2})$	pure killed process
Alon, Matias, and Szegedy [1] (L_2)	$\lvert x \rvert^2$	$O(\epsilon^{-2})$	Brownian motion
Indyk [17] (L_α)	$\lvert x \rvert^\alpha$	$O(\epsilon^{-2})$	α-stable process
Kane, Nelson, and Woodruff [20] (L_0)	$\mathbb{1}(P \mid x)$	$O(\epsilon^{-2} \log n)$	compound Poisson process over $\mathbb{Z}_P$, P random prime
Ganguly, Bansal, and Dube [16] ($L_{p,q}$)	$(\sum_{j \in [d]} \lvert x_j \rvert^p)^q$	$O(\epsilon^{-2})$	per-entry p-stable processes subordinated by per-row q-stable subordinators
Lévy-Tower **(new)**	f_X	$O(\epsilon^{-2} \log n)$	any Lévy process X over $\mathbb{R}^d$
Lévy-Stable **(new)**	f_X	$O(\epsilon^{-2})$	any stable Lévy process X over $\mathbb{R}^d$

Heterogeneous moments

It will become clear in the analysis that both Lévy-Tower and Lévy-Stable will work for *heterogeneous moments* $\sum_{v \in [n]} f_v(\mathbf{x}(v))$, where each element v can have its own function f_v. One only needs to simulate v's process $X^{(v)}$ according to f_v. For Lévy-Tower, f_v can be the characteristic exponent of any Lévy process. For Lévy-Stable with parameter $\alpha \in (0, 2]$, f_v can be the characteristic exponent of any α-stable Lévy process. For example, with $d = 2$ there are many 2-dimensional 1-stable processes to choose from, and one could have $f_1(x, y) = |x| + |y|$, $f_2(x, y) = |x + y|$, $f_3(x, y) = \sqrt{x^2 + y^2}$, each of which corresponds to a 2-dimensional 1-stable process.

1.5 Organization

We first review Lévy processes and the Lévy-Khintchine representation theorem in Section 2. Then present the Lévy-Tower and Lévy-Stable sketches in Section 3. In Section 4, we discuss the connection between previous sketches and Lévy processes in a greater detail (see Table 2 for a summary). In Section 5, we discuss the problem of characterizing the set of *tractable* functions, and describe the *Fourier-Hahn-Lévy* method for expanding the range of the Lévy-Tower beyond Lévy-Khintchine-representable functions. We conclude in Section 6 with a conjectured characterization of the class of tractable functions.

2 Preliminary: Lévy Processes

We use Sato's text [29] as our reference for the theory of Lévy processes.

2.1 Lévy Processes on $\mathbb{R}^d$

▶ **Definition 5** (Lévy processes [29, page 3]). *A random process $X = (X_t)_{t\in\mathbb{R}_+}$ on $\mathbb{R}^d$ is a* Lévy process *if it satisfies*

Stationary Increments. $X_{t+s} - X_t \sim X_s$ *for all* $t, s \in \mathbb{R}_+$.

Independent Increments. *for* $0 \le t_1 < t_2 \ldots < t_k$, $X_{t_1}, X_{t_2} - X_{t_1}, \ldots, X_{t_k} - X_{t_{k-1}}$ *are mutually independent*

Stochastic continuity. $X_0 = 0$ *almost surely and* $\lim_{t\searrow 0} \mathbb{P}(|X_t| > \epsilon) = 0$ *for any* $\epsilon > 0$.

▶ Remark 6. A process that satisfies stationary and independent increments is called *memoryless*.

The primary way to study Lévy processes is through their *characteristic functions*. See Table 1 for notations.

▶ **Theorem 7** (Lévy-Khintchine representation [29, page 37]). *Any Lévy process $X = (X_t)_{t\in\mathbb{R}_+}$ on $\mathbb{R}^d$ can be identified by a triplet (A, ν, γ) where A is a covariance matrix, ν is a measure on $\mathbb{R}$ such that*

$$\nu(\{0\}) = 0 \quad \text{and} \quad \int_{\mathbb{R}^d} \min(|s|^2, 1)\, \nu(ds) < \infty, \tag{1}$$

and $\gamma \in \mathbb{R}^d$. The identification is through the characteristic function. For any $t \in \mathbb{R}_+$ and $z \in \mathbb{R}^d$,

$$\mathbb{E}e^{i\langle X_t, z\rangle} = \exp\left(-t\left(\frac{1}{2}\langle z, Az\rangle - i\langle \gamma, z\rangle + \int_{\mathbb{R}^d} (1 + i\langle z, s\rangle \mathbb{1}(|s| < 1) - e^{i\langle z,s\rangle})\, \nu(ds)\right)\right). \tag{2}$$

Conversely, any triplet (A, ν, γ) where A is a covariance matrix, ν satisfies (1), and $\gamma \in \mathbb{R}^d$ corresponds to a Lévy process satisfying (2).

▶ Remark 8. In the one-dimensional case, the matrix A can be identified by the usual variance σ^2. Then we have for $z \in \mathbb{R}$,

$$\mathbb{E}e^{izX_t} = \exp\left(-t\left(\frac{1}{2}\sigma^2 z^2 - i\gamma z + \int_{\mathbb{R}} (1 + izs\mathbb{1}(|s| < 1) - e^{izs})\, \nu(ds)\right)\right). \tag{3}$$

We call the exponent in (2) the *characteristic exponent*, denoted by $f_X(z) = -\log \mathbb{E}e^{i\langle X_1, z\rangle}$. Let us note some properties of the characteristic exponent f.

▶ **Lemma 9.** *Let X be any Lévy process on $\mathbb{R}^d$ and f be its characteristic exponent.*

- *$\mathfrak{Re} f \ge 0$. $\mathfrak{Re} f = 0$ if and only if X is a deterministic drift, i.e., $X_t = \gamma t$ for all $t \in \mathbb{R}_+$.*
- *For any $x \in \mathbb{R}^d$, $f(-x) = \overline{f(x)}$, the complex conjugate of $f(x)$.*

We list some common one-dimensional Lévy processes in Table 3 that will be frequently used later.

Table 3 Common one-dimensional Lévy processes.

Process	Characteristic Exponent	Notes
linear drift	$i\gamma z$	$\gamma \in \mathbb{R}_+$
α-stable	$\lvert z\rvert^\alpha$	$\alpha \in (0, 2]$
Poisson	$1 - e^{iz}$	
compound Poisson	$\int_{-\infty}^{\infty}(1 - e^{izs})\,\nu(ds)$	$\nu(\mathbb{R}) < \infty$ is the jump rate, $\nu(\mathbb{R})^{-1}\nu$ is the jump distribution
symmetric compound Poisson	$2\int_0^{\infty}(1 - \cos(zs))\,\nu(ds)$	$\nu(\mathbb{R}) < \infty$

3 Infinitely Divisible Sketches

3.1 Lévy-Tower Sketches

We now present a sketch that is induced by a *generic* Lévy process on $\mathbb{R}^d$, which consists of two parameters.

- An accuracy parameter $m \in \mathbb{Z}_+$ which corresponds to the number of subsketches in classic settings.
- A Lévy process $X = (X_t)_{t\in\mathbb{R}_+}$ on $\mathbb{R}^d$ with characteristic exponent $f(z) = -\log \mathbb{E}e^{i\langle X_1, z\rangle}$.

▶ **Definition 10** ((f, m)-Lévy-Tower)**.** *Let f be the characteristic exponent of a Lévy process X on $\mathbb{R}^d$ and $m \in \mathbb{Z}_+$. An (f, m)-*Lévy-Tower *is an infinite vector $S = (S_k^{(j)})_{k\in\mathbb{Z}, j\in[m]} \subset \mathbb{T}^{\mathbb{Z}}$, initialized as all zero. For any element $v \in [n]$ and $y \in \mathbb{R}^d$, a vector update $\mathbf{x}(v) \leftarrow \mathbf{x}(v) + y$ is effected by:*

Update(v, y) **:** *For each $k \in \mathbb{Z}$ and $j \in [m]$, $S_k^{(j)} \leftarrow S_k^{(j)} + \left\langle y, X_{2^{-k}}^{(v,j)}\right\rangle \bmod 2\pi$, where $X^{(v,j)} = (X_t^{(v,j)})_{t\in\mathbb{R}_+}$ is an i.i.d. copy of the Lévy process X with characteristic exponent f.*

▶ Remark 11. If $|f(\mathbf{x})|$ is in the range $[1, \mathrm{poly}(n)]$, then it suffices to store levels $k \in [0, O(\log n)]$. Therefore a Lévy-Tower takes $O(\log n)$ words of space and each word stores a number in $[0, 2\pi)$.

The construction is standard in sketch design: subsample, randomly project, and then sum, but with an important and powerful tweak that the two steps "subsample and random project" are replaced by a single step: sampling at time 2^{-k} of the Lévy process whose characteristic exponent is f. Consequently, such sketches can be used to estimate f-moments. We characterize the distribution of the sketch as follows.

▶ **Lemma 12.** *Fix a frequency vector $\mathbf{x} = (\mathbf{x}(v))_{v\in[n]}$. For any $k \in \mathbb{Z}$, we have*

$$S_k \sim \sum_{v\in[n]} \left\langle X_{2^{-k}}^{(v)}, \mathbf{x}(v)\right\rangle$$

$$\mathbb{E}e^{iS_k} = e^{-2^{-k}f(\mathbf{x})}.$$

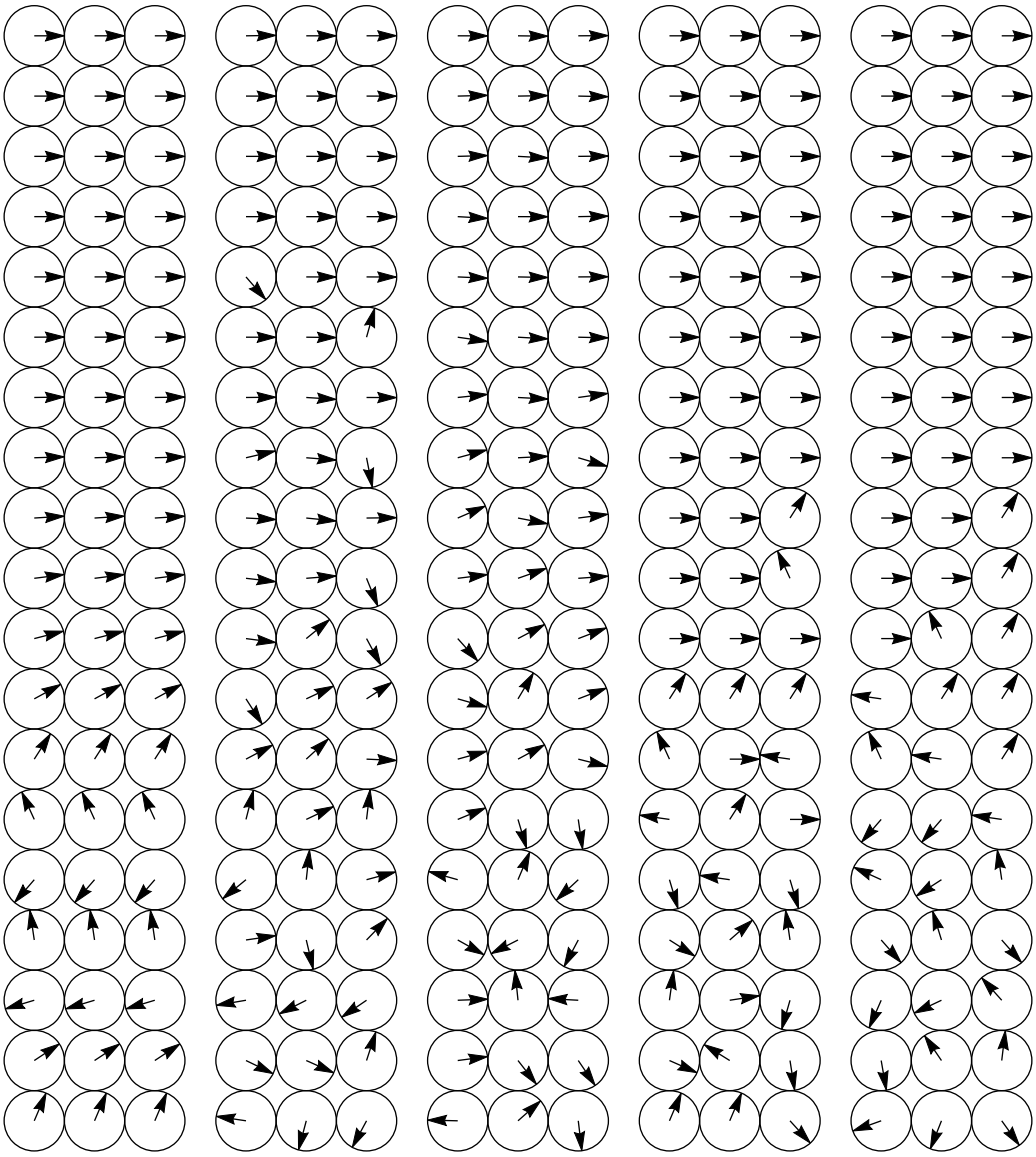


Figure 1 Lévy-Tower with $m = 3$ and $\mathbf{x} = (1, 0, \ldots, 0)$. From left to right: linear drift, Cauchy process, Brownian motion, Poisson process with rate 1, Poisson process with rate 2. Different Lévy processes have different "sensitivities" for a target function-moment. For example, linear drift is only sensitive to the sum of the vector and insensitive to how the values are distributed. The Cauchy process is only sensitive to the L_1-moment, while Brownian motion is only sensitive to the L_2-moment. Poisson processes are sensitive to the *support size* of $\mathbf{x}$ and at the same time leaking information about other f-moments.

Proof. The first statement is trivially true since the sketch is linear. For the second, note that the $X^{(v)}$ are i.i.d., and we have

$$\begin{aligned}
\mathbb{E}e^{iS_k} &= \mathbb{E}e^{i\sum_{v\in[n]}\left\langle X^{(v)}_{2^{-k}},\mathbf{x}(v)\right\rangle} \\
&= \prod_{v\in[n]} \mathbb{E}e^{i\left\langle X^{(v)}_{2^{-k}},\mathbf{x}(v)\right\rangle} && \text{(by independence)} \\
&= \prod_{v\in[n]} e^{-2^{-k}f(\mathbf{x}(v))} && \text{(by Lévy-Khintchine)} \\
&= e^{-2^{-k}f(\mathbf{x})}. && \text{(definition of } f(\mathbf{x})\text{)}
\end{aligned}$$

◀

Note that the f-moment lies exactly in the exponent of $\mathbb{E}e^{iS_k}$. We now formally present the estimation method.

3.2 Estimation of Lévy-Tower

As discussed in Section 1.2, the estimator needs to find a suitable level t and infer the f-moment from C_t. In the Lévy-Tower, we store $S_k = C_{2^{-k}}$ for all $k \in \mathbb{Z}$. We first prove the concentration of the empirical mean at each level.

▶ **Lemma 13.** *For any $t \in \mathbb{R}_+$,*

$$\mathbb{P}\left(\left|\frac{1}{m}\sum_{j=1}^{m} e^{iC_t^{(j)}} - e^{-tf(\mathbf{x})}\right| > \eta\right) \le \frac{2t|f(\mathbf{x})|}{m\eta^2}.$$

Proof. By Chebyshev's inequality, we have

$$\mathbb{P}\left(\left|\frac{1}{m}\sum_{j=1}^{m} e^{iC_t^{(j)}} - e^{-tf(\mathbf{x})}\right| > \eta\right) \le \frac{\mathbb{V}e^{iC_t}}{m\eta^2}.$$

Furthermore, since $\mathbb{E}e^{iC_t} = e^{-tf(\mathbf{x})}$, we have

$$\mathbb{V}e^{iC_t} = \mathbb{E}|e^{iC_t}|^2 - |\mathbb{E}e^{iC_t}|^2 = 1 - e^{-2t\mathfrak{Re}f(\mathbf{x})} \le 2t\mathfrak{Re}f(\mathbf{x}) \le 2t|f(\mathbf{x})|.$$ ◀

▶ **Lemma 14.** *With probability at least 99/100, for any $k \ge \log_2|f(\mathbf{x})|$, we have*

$$\left|\frac{1}{m}\sum_{j=1}^{m} e^{iS_k^{(j)}} - e^{-tf(\mathbf{x})}\right| \le O(1/\sqrt{m}).$$

Proof. By the union bound and Lemma 13,

$$\begin{aligned}
&\mathbb{P}\left(\exists k \ge \log_2|f(\mathbf{x})|, \left|\frac{1}{m}\sum_{j=1}^{m} e^{iS_k^{(j)}} - e^{-tf(\mathbf{x})}\right| > \eta\right) \\
&\le \sum_{k=\log_2|f(\mathbf{x})|}^{\infty} \frac{2\cdot 2^{-k}|f(\mathbf{x})|}{m\eta^2} \\
&= O\left(\frac{1}{m\eta^2}\right).
\end{aligned}$$

Thus it suffices to choose $\eta = O(1/\sqrt{m})$ for the probability above at most $1/100$. ◀

We consider the complex logarithm defined on $\mathbb{C}$ except for the negative real line.

▶ **Lemma 15.** *Let $x, y \in \{z \in \mathbb{C} : |1 - z| < 1/2\}$, then $|\log x - \log y| < 2|x - y|$.*

▶ **Theorem 16.** *With probability at least 99/100, the following procedure returns an estimate $\widehat{f(\mathbf{x})}$ such that $|\widehat{f(\mathbf{x})} - f(\mathbf{x})| \le O(|f(\mathbf{x})|/\sqrt{m})$. Let K be the maximum level and assume $K \ge \log_2 |f(\mathbf{x})| \ge 0$.*

- *Enumerate $k = K, K-1, \ldots, 0$ in the decreasing order, let $Y_k = \frac{1}{m}\sum_{j=1}^{m} e^{iS_k^{(j)}}$.*
  - *If $|1 - Y_k| > 0.2$, then return $-2^k \log Y_k$ and halt.*

Proof. By Lemma 14, with probability at least 99/100, we have $\left|Y_k - e^{-tf(\mathbf{x})}\right| \le O(1/\sqrt{m})$ for all $k \in [\log_2 |f(\mathbf{x})|, K]$. We assume this event happens. Let W be the index selected in the procedure. First note that for $z \in \mathbb{C}$ with $\mathfrak{Re} z \ge 0$, one has $|1 - e^{-z}| \le |z|$. In addition, by Lemma 9, we know $\mathfrak{Re} f(\mathbf{x}) \ge 0$. Thus

$$|1 - Y_k| = |1 - e^{-tf(\mathbf{x})}| + O(1/\sqrt{m}) \le t|f(\mathbf{x})| + O(1/\sqrt{m}).$$

Let m be large enough that $O(1/\sqrt{m}) < 0.01$. Since the times are base 2, one has

$$\begin{aligned} 2^{-W} &\in [(0.2 - 0.01)/|f(\mathbf{x})|, 2(0.2 + 0.01)/|f(\mathbf{x})|] \\ &\subset (0.2/|f(\mathbf{x}))|, 0.45/|f(\mathbf{x}))|). \end{aligned}$$

Thus we have $|1 - Y_W| < 0.45 + O(1/\sqrt{m}) < 0.5$. By Lemma 15, we have

$$|-\log Y_W - 2^{-W} f(\mathbf{x})| \le 2|Y_W - e^{-2^{-W} f(\mathbf{x})}| \le O(1/\sqrt{m}).$$

We conclude that

$$|-2^W \log Y_W - f(\mathbf{x})| \le O(2^W/\sqrt{m}) = O(|f(\mathbf{x})|/\sqrt{m}). \qquad \blacktriangleleft$$

3.3 Lévy-Stable Sketches

Recall that a d-dimensional Lévy process X is *α-stable* if for any $t \in \mathbb{R}_+$, $X_t \sim t^{1/\alpha} X_1$. We call the characteristic exponents of stable processes *stable moments*. Stable moments are of special interest in the context of streaming sketches since there is no need to store the whole tower; only m i.i.d. samples suffice to return an estimate with relative error $O(1/\sqrt{m})$.[8] The only symmetric one-dimensional stable processes are α-stable random processes, which correspond to Indyk's sketches [17]. On the other hand, there is a rich class of higher dimensional stable processes, the one implicit in Ganguly et al. [16] for estimating $L_{p,q}$ hybrid moments being just a single special case.

▶ **Theorem 17** (Lévy-Khintchine for stable processes [29, page 86])**.** *Let X be a Lévy process on $\mathbb{R}^d$ and $x \in \mathbb{R}^d$. X is 2-stable if and only if $\mathbb{E}e^{i\langle x, X_t\rangle} = e^{-\frac{1}{2}t\langle x, Ax\rangle}$ for some covariance matrix A. If X is symmetric, then it is α-stable for $\alpha \in (0, 2)$ if and only if there is a finite, positive measure μ on the sphere $\mathbb{S}_{d-1} = \{x \in \mathbb{R}^d : |x| = 1\}$ such that*

$$\mathbb{E}e^{i\langle x, X_t\rangle} = \exp\left(-t\int_{\mathbb{S}_{d-1}} |\langle x, \xi\rangle|^\alpha \, \mu(d\xi)\right).$$

[8] As discussed in Section 1.3, the Lévy-Towers induced by stable processes are *self-similar*: all registers in the tower are identically distributed with a proper normalization.

▶ Remark 18. For simplicity we only consider symmetric processes here when $\alpha \in (0,2)$ so that the characteristic exponent $f_X(x_1,\ldots,x_d) = f_X(|x_1|,\ldots,|x_d|)$ for any $x \in \mathbb{R}^d$. See [29] for the full characterization.

As discussed in Section 1.3, any multidimensional α-stable moment is as simple to estimate as the one-dimensional L_α-moment, since it is possible to maintain a random variable distributed as $f(\mathbf{x})^{1/\alpha}Y_\alpha$ where Y_α is a unit α-stable random variable. We now give the formal definition of the Lévy-Stable sketch for completeness.

▶ **Definition 19** ((f,m)-Lévy-Stable). *Let f be the characteristic exponent of an α-stable Lévy process X on $\mathbb{R}^d$ and $m \in \mathbb{Z}_+$. An (f,m)-Lévy-Stable sketch is a vector of m registers $T = (T_1,\ldots,T_m) \in \mathbb{R}^m$, initialized as all zero. For any element $v \in [n]$ and $y \in \mathbb{R}^d$, the vector update $\mathbf{x}(v) \leftarrow \mathbf{x}(v) + y$ is effected by:*

Update(v,y) **:** *For each $k \in [m]$, $T_k \leftarrow T_k + \left\langle y, X_k^{(v)} - X_{k-1}^{(v)} \right\rangle$, where $X^{(v)} = (X_t^{(v)})_{t\in\mathbb{R}_+}$ is an independent copy of the Lévy process X with characteristic exponent f, indexed by v.*

▶ Remark 20. By construction, the Lévy-Stable sketches emulates Indyk's sketch, storing m words where each word stores a real number of magnitude at most $\mathrm{poly}(n)$ with high probability, given $f(\mathbf{x}) \le \mathrm{poly}(n)$.

▶ Remark 21. We remark here on the differences between the Lévy-Tower sketch and the Lévy-Stable sketch.

- Lévy-Tower estimates generic function moment f_X where X is the characteristic exponent of *any* Lévy process. Lévy-Tower stores projections at multiple times of the process and each projection is wrapped around within $[0,2\pi)$.
- Lévy-Stable estimates function moment f_X where X is the characteristic exponent of a *stable* Lévy process. Lévy-Stable stores projections at unit time of the process and each projection is in $\mathbb{R}$. Lévy-Stable generalizes the stable sketches of Indyk [17] and Ganguly et al. [16].

4 Previous Sketches and Lévy Processes

In this section, we will discuss in greater detail how previous sketching techniques are equivalent to sampling Lévy processes. The guiding question is: what will happen if the input vector is repeated a large number of times on disjoint supports?

4.1 Single-Level Aggregation and the Central Limit Theorem

We start from the AMS sketch by Alon, Matias, and Szegedy [1], where a cell Q stores

$$Q = \sum_{v\in[n]} \mathbf{x}(v)\xi_v.$$

Here $\mathbf{x}$ is a $\mathbb{Z}$-stream and the $\xi_v \in \{-1,1\}$ are i.i.d. Rademacher random variables.[9] Now suppose the input vector is repeated w times on disjoint supports. Then the final state would be

$$M_w = Q_1 + Q_2 + \cdots + Q_w,$$

[9] Again, in [1]'s analysis, 4-wise independence hashing suffices to guarantee a good enough estimate but we need to assume they are i.i.d. here to talk about the exact distribution of the final state.

where the (Q_j) are i.i.d. copies of Q. Note that $\mathbb{E}Q = 0$ and $\mathbb{V}Q = \sum_{v\in[n]} \mathbf{x}(v)^2$. Thus as $w \to \infty$ the normalized final state $M_w/\sqrt{w}$ converges to a centered Gaussian random variable with variance $\sum_{v\in[n]} \mathbf{x}(v)^2$ in distribution, by the central limit theorem. Thus, in the regime as $w \to \infty$, the AMS sketch eventually becomes the same as a cell in the Lévy-Tower $\sum_{v\in[n]} \left\langle X_1^{(v)}, \mathbf{x}(v) \right\rangle$, where the $(X^{(v)})$ are i.i.d. one dimensional Brownian motion and $X_1^{(v)}$ is the value of the process at time $t = 1$. Of course, since Brownian motions are self-similar, one does not need to sample the Lévy processes at exponentially spaced intervals; samples from m independent processes at time $t = 1$ suffice to approximate $\sum_{v\in[n]} \mathbf{x}(v)^2$ with $O(1/m)$ relative variance. The characteristic exponent of Brownian motion is the target function for the L_2-moment: $f(x) = |x|^2$.

Another illuminating example is Ganguly's [15] L_k-moment estimator for $k \geq 3$. Ganguly randomly projects the elements and stores $S = \sum_{v\in[n]} \mathbf{x}(v)Z^{(v)}$ where each $Z^{(v)}$ is a random root of $x^k = 1$ on the complex unit circle. The statistic S^k seems to be a good estimator for F_k-moment since $\mathbb{E}S^k = \sum_{v\in[n]} \mathbf{x}(v)^k$. However, since this random projection has finite variance, the normalized sum $S/\sqrt{\sum_{v\in[n]} |\mathbf{x}(v)|^2}$ will converge to a complex Gaussian as the input vector is duplicated on disjoint supports. In the limit, the only information remaining in the sketch pertains to the L_2-moment. Indeed, one in fact needs the number of i.i.d. registers to grow polynomially in the support-size for it to work for F_k [15].

By the generalized central limit theorem, for whatever random variable Q that is produced by the current input $\mathbf{x}$, if for some sequences $(a_w)_{w\in\mathbb{N}}$ and $(b_w)_{w\in\mathbb{N}}$, $(M_w - b_w)/a_w$ converges to some non-degenerate random variable Y as $w \to \infty$, then Y has to be α-stable for $\alpha \in (0, 2]$. If in addition, $\mathbb{V}Q < \infty$, then Y is Gaussian. Non-Gaussian stable distributions are discussed later in Section 4.3.

4.2 Multi-Level Subsampling and the Poisson Limit Theorem

Another important sketching technique is *subsampling*. The idea is to devise a sketch that works for $\Theta(m)$ elements and then solve the generic case by subsampling the stream at rates 2^{-k} for $k \in \mathbb{N}$, one of which reduces it down to $\Theta(m)$ elements. Without loss of generality, suppose now we have m non-zero elements with distinct values $\mathbf{x}(1), \ldots, \mathbf{x}(m)$. Let w be the number of times we repeat the input stream and denote the repeated stream as $\mathbf{x}^w$. To obtain a $\Theta(m)$-size set of subsamples, one needs to subsample $\mathbf{x}^w$ with rate $1/w$. Let $Y_{w,j}$ be the indicator that the jth copy of $\mathbf{x}(1)$ is sampled where $\mathbb{E}Y_{w,j} = 1/w$. The number of elements with value $\mathbf{x}(1)$ in the subsampled set is

$$\sum_{j=1}^{w} Y_{w,j} \to \text{Poisson}(1) \text{ in distribution as } w \to \infty,$$

where we have invoked the Poisson limit theorem (see, e.g., [12, Theorem 3.6.1]) because $\mathbb{E}\sum_{j=1}^{w} Y_{w,j} = 1$ and $\max_{j=1}^{w} \mathbb{P}(Y_{w,j} \neq 0) = \frac{1}{w} \to 0$ as $w \to \infty$. Similarly, the number of elements with value $\mathbf{x}(j)$ is also Poisson(1) and the occurrences of different values are independent. It is well known that such a limiting distribution can be simulated algorithmically by duplicating every element a Poisson(1) number of times (see [14, 13, 27, 30]). On the other hand, such limit distributions can be equivalently simulated by hashing each element to a Poisson process and then sampling at different times. Thus all sketches based on subsampling can be simulated by the Lévy-Tower (without mapping to the complex unit circle $\mathbb{T}$) with the corresponding (compound) Poisson processes.

4.3 Stable Random Variables and Stable Processes

Whenever Q has a finite variance, $(Q_1 + Q_2 + \ldots + Q_w)/\sqrt{w}$ goes to Gaussian as $w \to \infty$. Nevertheless, with infinite variance, Q can lie in the domain of attraction of an α-stable distribution for any $\alpha \in (0, 2)$. Indeed, the use of stable random variables is another sketching technique that directly corresponds to Lévy processes: α-stable processes. Similar to the Gaussian case (Section 4.1), α-stable processes are self-similar so there is no need to store the whole tower with exponentially spaced sample times 2^{-k}. Rather, it suffices to sample m independent processes at time $t = 1$.

Indyk [17] uses one-dimensional α-stable random variables for $\alpha \in (0, 2]$ to estimate the L_p-moment, and indeed, the characteristic exponent of the α-stable process is $f(x) = |x|^\alpha$. Ganguly, Bansal, and Dube [16] consider the higher dimensional case, where each element has d attributes. We now show how to reconstruct [16]'s sketch from the perspective of Lévy processes. The target function is

$$f : \mathbb{R}^d \to \mathbb{C}, \quad f(x) = \left(\sum_{j=1}^{d} |x_j|^p \right)^q.$$

Such f-moments are called $L_{p,q}$ hybrid moments in [16]. By the subordination theorem [29, page 197], this function is the characteristic exponent of a vector of d independent p-stable processes that are subordinated by a common q-stable subordinator. That is,

$$X_t = (X^{[1]}_{Z_t}, \ldots, X^{[d]}_{Z_t}),$$

where Z is a q-stable subordinator and the $(X^{[j]})$ are i.i.d. p-stable processes. Note that since $X^{[j]}$ is p-stable, we have $X^{[j]}_{Z_t} \sim Z_t^{1/p} X^{[j]}_1$. Thus, we sample X at time 1 and get

$$X_1 \sim Z_1^{1/p} \left(X^{[1]}_1, \ldots, X^{[d]}_1 \right).$$

Take the inner product and we have, for $x \in \mathbb{R}^d$,

$$\langle x, X_1 \rangle = Z_1^{1/p} \sum_{j=1}^{d} x_j X^{[j]}_1,$$

which is exactly the random projection defined in [16].

4.4 HyperLogLog, PCSA, and Pure Killed Processes

Cardinality sketches like HyperLogLog [13] and PCSA [14] that only allow increments also correspond to Lévy processes in a surprisingly natural way. We consider the number system $\mathbb{R} \cup \{\infty\}$, where for any $x \in \mathbb{R}$, $x + \infty = \infty + x = \infty$ and $\infty + \infty = \infty$. Such definitions extend to multiplication with natural numbers where $0 \cdot \infty = 0$ and $k \cdot \infty = \infty$ for any $k \in \mathbb{Z}_+$.

▶ **Definition 22** (pure killed processes). *A* pure killed process $X = (X_t)_{t \in \mathbb{R}_+}$ *with* kill rate $c > 0$ *is a Lévy process over* $\mathbb{R} \cup \{\infty\}$ *which can be simulated as follows.*

- *Sample a* kill time $Y \sim \mathrm{Exp}(c)$.
- $X_t = 0$ *if* $t < Y$ *and* $X_t = \infty$ *otherwise.*

In particular, we have $\mathbb{P}(X_t = \infty) = \mathbb{P}(Y \le t) = 1 - e^{-ct}$.

Assume the insertion stream is $v_1, \ldots, v_T$, where each v_j is an element in the universe $[n]$. Let $X = (X_t)_{t\in\mathbb{R}_+}$ be a pure killed process with unit kill rate ($c = 1$). Similar to the turnstile case in Section 1.2, one stores the sum at time t,

$$C_t = \sum_{j=1}^{T} X_t^{(v_j)} = \sum_{v\in[n]} \mathbf{x}(v) \cdot X_t^{(v)},$$

where $X^{(v)}$s are i.i.d. copies of X. It is straightforward to see that S_k perfectly simulates a Poissonized PCSA cell [14, 27] (Table 4).

Table 4 HyperLogLog and PCSA can be considered as Lévy-Tower sketches with pure killed processes where the bit-or operation $(\{0,1\}, \vee)$ is simulated with the extended real numbers $(\{0,\infty\}, +)$. Note that, of course, such a reinterpretation does not upgrade HyperLogLog and PCSA to work over turnstile streams. The resulting Lévy-Tower is still incremental-only since $\infty - \infty$ is not defined.

bit operation in cardinality sketches	jumps of a pure killed process
$0 \vee 0 = 0$	$0 + 0 = 0$
$0 \vee 1 = 1$	$0 + \infty = \infty$
$1 \vee 0 = 1$	$\infty + 0 = \infty$
$1 \vee 1 = 1$	$\infty + \infty = \infty$

In fact, one computes

$$\mathbb{E}e^{iC_t} = \prod_{v\in[n]} \mathbb{E}e^{i\mathbf{x}(v)\cdot X_t^{(v)}} = \left(\mathbb{E}e^{iX_t}\right)^{\|\mathbf{x}\|_0} \tag{4}$$

since $0 + \cdots + 0 = 0$ and $\infty + \cdots + \infty = \infty$. By Definition 22, $X_t = 0$ with probability e^{-t}, hence $e^{iX_t} = 1$, and $X_t = \infty$ with probability $1 - e^{-t}$, in which case $e^{iX_t} = 0$.[10] Thus we have $\mathbb{E}e^{iX_t} = e^{-t}$. Inserting this back to Equation (4), we have

$$\mathbb{E}e^{iC_t} = e^{-t\|\mathbf{x}\|_0}.$$

This coincides with the observation in Section 1.2, where the estimation target $\|\mathbf{x}\|_0$ lies right in the exponent of $\mathbb{E}e^{iC_t}$.

We note that HyperLogLog and PCSA both correspond to the pure killed process, with different couplings of cells on different levels. In particular, HyperLogLog stores $(C_1, C_{1/2}, C_{1/4}, \ldots)$ where $C_{2^{-j}}$s are sampled from *a single process* (exactly as in the Lévy-Tower) while PCSA stores $(C_1, C_{1/2}, C_{1/4}, \ldots)$ where $C_{2^{-j}}$s are independent. Since a pure killed process will remain at ∞ after its first jump, for HyperLogLog, $(C_1, C_{1/2}, C_{1/4}, \ldots)$ will always be in the form of a prefix of ones (∞s) followed by all zeros. Therefore, HyperLogLog only needs to store the length of the prefix of ones.

4.5 Uniform Random Projection and the Integral Lévy-Tower

Indyk's L_p-stable sketches [17] are able to estimate the L_p-moment for $p \in (0, 2]$ but do not handle the L_0-moment. Cormode, Datar, Indyk, and Muthukrishnan [10] approximate the L_0-moment by the L_α-moment for very small $\alpha > 0$. This scheme needs a bound on $\|\mathbf{x}\|_\infty$, as a single unbounded update can raise the estimate to infinity. Kane, Nelson, and

[10] Here we take the Cesàro limit $e^{i\infty} = \lim_{T\to\infty} \frac{1}{T}\int_0^T e^{it}\,dt = 0$.

Woodruff [20] take another approach: subsample the stream to a suitable level, randomly project each element within the field $\mathbb{Z}_p$ for a random prime $p > \epsilon^{-1}\log(\|\mathbf{x}\|_\infty)$, and then use an (insertion-only) F_0 estimator. The random projection approach [20] also requires a bound on $\|\mathbf{x}\|_\infty$. As an L_0-moment sketch it has a certain failure probability (e.g., $p|\mathbf{x}(v)$ for many v) but it perfectly estimates a related quantity: the $f_{L_0,p}$-moment for $f_{L_0,p}(x) = \mathbb{1}(p \nmid x)$.

We show how the Lévy-Tower reconstructs [20]'s *trick* by *pure computation*. The target function $f(k) = \mathbb{1}(p \nmid k)$ for $k \in \mathbb{Z}$ can be written as

$$f(x) = \frac{1}{p}\sum_{j=0}^{p-1}(1 - \cos(2\pi jx/p)).$$

The Lévy process with characteristic exponent f is the compound Poisson process with jumps uniformly chosen from $-2\pi(p-1)/p, \ldots, 0, \ldots, 2\pi(p-1)/p$. See Figure 2. The corresponding Lévy-Tower reconstructs [20]'s sketch: subsample through Poisson processes and uniformly project over $\mathbb{Z}_p$ with the uniform random jumps. It is straightforward to see that the resulting Lévy-Tower is *integral* in the sense that $S_k \in \{2j\pi/p : j = 0, \ldots, p\}$ for any k and S_k can be identified by a $\mathbb{Z}_p$-value. Moreover, while [20]'s estimator needs to assume p is a prime, the Lévy-Tower will work no matter whether p is prime or not.

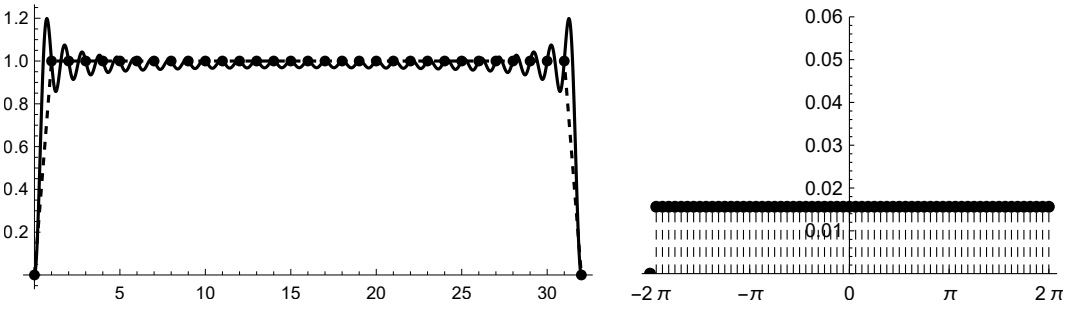


Figure 2 Left: $f_{L_0,32}(x) = \sum_{j=1}^{31}\frac{1}{32}(1-\cos(2\pi jx/32))$. The black dots mark the values at integer xs. Right: The jump distribution of the compound Poisson process X with characteristic exponent $f_{L_0,32}$. The subsampling and uniform random projection tricks used in [20] are recovered from computing the corresponding Lévy process.

5 Tractability and Lévy Processes

A function $f : \mathbb{Z} \to \mathbb{R}$ is *tractable* [4] if the f-moment of any input stream $\mathbf{x} \in \{-M, \ldots, M\}^n$, where $M = O(\text{poly}(n))$, can be $(1 \pm \epsilon)$-approximated with $O(\text{poly}(\epsilon^{-1}, \log n))$ space. Monotonic functions are fully characterized in [6] and all functions except for a class of nearly periodic functions are characterized in [4]. The Lévy-Tower tracks every function f that is the characteristic exponent of a Lévy process, and such exponents are characterized by Lévy-Khintchine. In this section, we investigate the set of tractable functions implied by the Lévy-Khintchine representation.

We start the discussion with a comparison between one-dimensional real characteristic exponents and existing communication complexity lower bounds. Note that by Lévy-Khintchine, such exponents can be written as $f(x) = \frac{1}{2}\sigma^2x^2 + 2\int_0^\infty(1-\cos(xs))\,\nu(ds)$.

▶ **Lemma 23.** *Let $f(x) = \frac{1}{2}\sigma^2x^2 + 2\int_0^\infty(1-\cos(xs))\,\nu(ds)$. The following statements are true.*

- *$f(x) \geq 0$ for any $x \in \mathbb{R}$. [Commentary: functions with both positive and negative values require $\Omega(\text{poly}(n))$-size sketches for constant factor approximations [9].]*

- *For any $z \in \mathbb{Z}$ and $x \in \mathbb{R}$, $f(zx) \le z^2 f(x)$. [Commentary: functions increasing faster than quadratic require $\Omega(\text{poly}(n))$-size sketches for constant factor approxiamtions [1, 2].]*
- *For any $z \in \mathbb{R}_+$, $f(z) = 0$ if and only if $f(x+z) = f(x), \forall x \in \mathbb{R}$, i.e., f is periodic with period z. [Commentary: functions that have zeros other than the origin require $\Omega(\text{poly}(n))$-size sketches, unless the function is periodic with period $\min\{z > 0 : f(z) = 0\}$ [9].]*

Proof. The first statement is obvious. For the second statement, it suffices to prove that with $y = xs$,

$$(1 - \cos(zy)) \le z^2(1 - \cos(y)),$$

for any $y \in \mathbb{R}, z \in \mathbb{Z}$. When $z = 0$, the statement trivially holds. Without loss of generality, assume $z \in \mathbb{Z}_+$ and let $g(y) = z^2(1 - \cos(y)) - (1 - \cos(zy))$. Check that $g'(y) = z^2 \sin(y) - z\sin(zy)$ and $g''(y) = z^2\cos(y) - z^2\cos(zy)$. Clearly for $y \in [0, \pi/z]$, we have $\cos(zy) \le \cos(y)$ and thus $g''(y) \ge 0$. Note $g'(0) = 0$ and therefore $g(y) \ge 0$ for $y \in [0, \pi/z]$. Now consider $y \in [\pi/z, \pi]$, $z^2(1 - \cos(y)) \ge z^2(1 - \cos(\pi/z)) \ge 2 \ge 1 - \cos(zy)$, which implies $g(y) \ge 0$ on $[0, \pi]$. Thus we have $g(y) \ge 0$ for $[\pi, 2\pi]$ too by symmetry. The statement is thus proved since $g(y)$ is periodic in 2π.

For the third statement, $f(z) = 0$ necessarily implies that $\sigma = 0$ and the measure ν concentrates on $2\pi j/z$ for $j \in \mathbb{Z}$. Thus $f(x) = \sum_{j\in\mathbb{Z}}(1 - \cos(2\pi jx/z))\nu(\{2\pi j/z\})$ is periodic with period z. ◀

Braverman, Chestnut, Woodruff, and Yang [4] characterize the tractability of all symmetric functions over $\mathbb{Z}$, except for a class of nearly periodic functions. Specifically, the following example is given in [4, §5] which cannot be solved in the L_2-heavy hitter-based framework, but can be estimated using *ad hoc* algorithmic tricks.

▶ **Definition 24** (g_{np}, a nearly periodic function [4, §5]). *For $x \in \mathbb{N}$, $g_{np}(x) = 2^{-\tau(x)}$, where $\tau(x) = \max\{j \in \mathbb{N} : 2^j | x\}$, i.e., the position of the first "1" in the binary representation of x.*

The reason that g_{np} cannot be tracked by finding L_2-heavy hitters is that $g_{np}(x)$ can occasionally becomes polynomially small in x. In particular, for $k \in \mathbb{N}$, $g_{np}(2^k) = 2^{-k}$. We now demonstrate how the function g_{np} can be estimated *directly* by the Lévy-Tower.

5.1 Estimating g_{np} with the Lévy-Tower

Since it is assumed the element values are at most $\text{poly}(n)$, it suffices to consider a 2^w-periodic version of g_{np} where $w = O(\log(n))$.

▶ **Definition 25** (2^w-periodic g_{np}). *For $x \in \mathbb{N}$,*

$$g_{np,w}(x) = \begin{cases} g_{np}(x), & 2^w \nmid x \\ 0, & 2^w \mid x. \end{cases}$$

We show that, indeed, g_{np} corresponds to a Lévy process, specifically a compound Poisson process. The proof is included in the full version.

▶ **Lemma 26.** *Fix $w \in \mathbb{N}$. For any $x \in \mathbb{N}$,*

$$g_{np,w}(x) = \sum_{j=1}^{2^w-1} \frac{2^{2\tau(j)+1} + 1}{3 \cdot 2^{2w-1}}(1 - \cos(2\pi jx/2^w)).$$

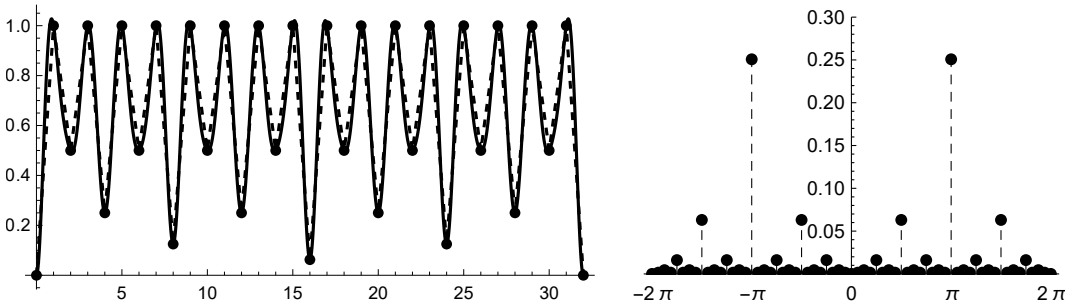


Figure 3 Left: $g_{np,5}(x) = \sum_{j=1}^{31} \frac{2^{2\tau(j)+1}+1}{1536}(1-\cos(2\pi jx/32))$. The black dots mark the values at integer xs. Right: The jump distribution of the compound Poisson process X with characteristic exponent $g_{np,5}$. Such nearly periodic functions do not fit in the L_2-heavy-hitter based framework in [4]. Nevertheless, one may compute the corresponding Lévy process and apply the Lévy-Tower.

The jump rate is

$$\sum_{j=1}^{2^w-1} \frac{2^{2\tau(j)+1}+1}{3\cdot 2^{2w-1}} = \frac{2^{2w}-1}{3\cdot 2^{2w-1}}.$$

By Lévy-Khintchine, $g_{np,w}$ is the characteristic function of a compound Poisson process X with jump rate $\frac{2^{2w}-1}{3\cdot 2^{2w-1}}$ and jump J distributed as

$$\mathbb{P}(J = 2\pi j/2^w) = \mathbb{P}(J = -2\pi j/2^w) = \frac{1}{2}\frac{2^{2\tau(j)+1}+1}{2^{2w}-1}, \quad \text{for } j = 1,\ldots 2^w-1.$$

Moreover, the parameter w for the random jump induced by $g_{np,w}$ has a neat algorithmic interpretation as the max recursion depth if one implements each random jump by a recursive algorithm[11]. The case where $w \to \infty$ corresponds to the algorithm without depth limit. Let $\mathbb{B}$ be the set of all real fractions ($\in [0,1)$) with finite binary representations. Note that the jump distribution converges pointwisely to

$$\mathbb{P}(J = 2\pi x) = \mathbb{P}(J = -2\pi x) = \frac{1}{2}2^{1-2\tau_*(x)},$$

for any $x \in \mathbb{B}$, where $\tau_*(x)$ returns the location of the first one in the binary representation. For example $\tau_*(1/4) = \tau_*((0.01)_2) = 2$. We may thus simulate such jumps by the following algorithm to handle unbounded input streams with dynamically allocated memory.

▶ **Lemma 27.** *Let A, B be two random bit-tapes, embedded on reals with the binary expansion ($A = \sum_{j=1}^{\infty} A[j]2^{-j}, B = \sum_{j=1}^{\infty} B[j]2^{-j}$), and let $\xi \in \{-1,1\}$ be a Rademacher random variable. Then*

$$J \sim 2\pi\,\xi\left(\sum_{j=1}^{\tau_*(B)-1} A[j]2^{-j} + 2^{-\tau_*(B)}\right).$$

[11] (Halt with prob. 1/2; go left and repeat with prob. 1/4; and go right and repeat with prob. 1/4; see Figure 3).

5.2 The Fourier-Hahn-Lévy Method

The example above shows that it is possible to systematically sketch nearly periodic functions by sketching periodic functions and letting the period go to infinity. A natural conjecture is that the class of tractable functions are precisely those with a Lévy-Khintchine representation. This conjecture is *absolutely false*. We refute it with a simple tractable function having no Lévy-Khintchine representation, then attempt to salvage the conjecture via what we call the *Fourier-Hahn-Lévy* method.

▶ **Example 28** (the 0-1-5 problem)**.** Consider the function f where $f(0) = 0$, $f(1) = 1$, $f(k) = 5$ for all $k \geq 2$. The f-moment is easy to estimate to within a $1 \pm \epsilon$ factor in $O(\epsilon^{-2} \operatorname{polylog}(n))$ space, using ℓ_0-sampling [11]. However, no Lévy process has its characteristic function equal to f, for otherwise we would have $f(2 \cdot 1) \leq 2^2 f(1)$ by Lemma 23, which implies $5 \leq 4$ and is absurd.

We now extend our method to handle the *0-1-5 problem*. To be more concrete, we consider an arbitrary symmetric, real function f defined on $\{-p+1, \dots 0, \dots, p-1\}$ where $p > \|\mathbf{x}\|_\infty$ is the range of the values, say 2^{64}. We now describe the three-step *Fourier-Hahn-Lévy method*.

Fourier Transform. Decompose $f(0) - f(x) = \sum_{j=0}^{p-1}(1 - \cos(2\pi xj/p))\hat{f}(j)$ where the coefficient $\hat{f}(j)$ can be computed by taking the Fourier transform of f. Note that $\hat{f}$ is real since we have assumed f to be real and symmetric.

Hahn Decomposition. Decompose $\hat{f} = \hat{f}_+ - \hat{f}_-$ where $\hat{f}_+$ and $\hat{f}_-$ are non-negative real functions.

Lévy Process Simulation. Define functions $f_{\widehat{+}}, f_{\widehat{-}}$ as the inverse Fourier transform of $\hat{f}_+, \hat{f}_-$ respectively. Since the Fourier transform is linear, we have $f = f_{\widehat{+}} - f_{\widehat{-}}$. By construction, $f_{\widehat{+}}, f_{\widehat{-}}$ are all characteristic exponents of some Lévy processes. Estimate those function moments separately with Lévy-Tower sketches and sum to yield an estimate of the f-moment.

▶ **Lemma 29.** *Given any symmetric function $f : \mathbb{Z} \to \mathbb{R}_+$ with $f(0) = 0$ and any stream $\mathbf{x} \in [-p+1, p-1]^n$, the resulting estimator $\widehat{f(\mathbf{x})}$ of the Fourier-Hahn-Lévy method has error*

$$\left|\widehat{f(\mathbf{x})} - f(\mathbf{x})\right| \leq O((|f_{\widehat{+}}(\mathbf{x})| + |f_{\widehat{-}}(\mathbf{x})|)/\sqrt{m}),$$

with probability at least 98/100.

If f is the characteristic exponent of a Lévy process, then $f = f_{\widehat{+}}$ and $f_{\widehat{-}} = 0$. This case was already considered, where we get an asymptotically unbiased estimate with multiplicative error. Clearly, whenever $|f_{\widehat{+}}(\mathbf{x})| + |f_{\widehat{-}}(\mathbf{x})| = O(|f(\mathbf{x})|)$, the Fourier-Hahn-Lévy method *still* returns an estimate with multiplicative error, which handles the *0-1-5 problem*, among others.

6 Conclusion

The design of efficient streaming sketches has traditionally been more *art* than *science*, in which every sketching problem seemed to require a new insight or trick. Some prominent examples are F_0-sketching ($\{0, \infty\}$ projection [14]), L_2-sketching (sub-gaussian projection [1]), L_p-sketching (stable projection [17]), and L_0-sketching ($\mathbb{Z}_p$ uniform projection [20]); see also [16, 13].[12]

[12] There are *universal sketches* [6, 4] which can estimate a large class of function moments with a single sketch. But they consume much more space compared to classic sketches and the ones presented in this paper.

In this work we provide a more systematic method for generating sketches to estimate f-moments, by explicitly using the corresponding Lévy process whose characteristic exponent is f. This scheme mechanically *recreates* many known sketches [14, 1, 17, 16] and provides prototypical solutions to estimating nearly periodic moments, multidimensional moments, and heterogeneous moments. The key idea is to leverage the Lévy-Khintchine representation theorem for Lévy processes in the *design* and *analysis* of sketching algorithms.

Are there any f-moments that are *tractable* [6, 4] (they can be estimated to within a constant factor with a small sketch) but *do not* correspond to Lévy processes? In Section 5 we showed that the answer to this narrow question is *yes*, then introduced the *Fourier-Hahn-Lévy method*, which reduces an f without a Lévy-Khintchine representation to the difference of two functions $f_+^\wedge - f_-^\wedge$ that *can* be estimated by the Lévy-Tower. Does the Fourier-Hahn-Lévy method capture all tractable f-moments? We conjecture the answer is yes.

▶ **Conjecture 30.** *If $f : \mathbb{Z} \to \mathbb{C}$ is tractable, then the f-moment can be estimated to within a constant factor with a Lévy-Tower, either directly or after applying the Fourier-Hahn-Lévy method.*

References

1 Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999. doi:10.1006/jcss.1997.1545.

2 Ziv Bar-Yossef, Thathachar S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004. doi:10.1016/J.JCSS.2003.11.006.

3 Vladimir Braverman and Stephen R Chestnut. Universal sketches for the frequency negative moments and other decreasing streaming sums. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, page 591, 2015.

4 Vladimir Braverman, Stephen R. Chestnut, David P. Woodruff, and Lin F. Yang. Streaming space complexity of nearly all functions of one variable on frequency vectors. In *Proceedings 35th ACM Symposium on Principles of Database Systems (PODS)*, pages 261–276, 2016. doi:10.1145/2902251.2902282.

5 Vladimir Braverman, Jonathan Katzman, Charles Seidell, and Gregory Vorsanger. An optimal algorithm for large frequency moments using $O(n^{1-2/k})$ bits. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2014)*. Schloss-Dagstuhl – Leibniz Zentrum für Informatik, 2014. doi:10.4230/LIPIcs.APPROX-RANDOM.2014.531.

6 Vladimir Braverman and Rafail Ostrovsky. Zero-one frequency laws. In *Proceedings 42nd ACM Symposium on Theory of Computing (STOC)*, pages 281–290, 2010. doi:10.1145/1806689.1806729.

7 Vladimir Braverman and Rafail Ostrovsky. Generalizing the layering method of Indyk and Woodruff: Recursive sketches for frequency-based vectors on streams. In *Proceedings 16th International Workshop on Approximation, Randomization, and Combinatorial Optimization (APPROX)*, volume 8096 of *Lecture Notes in Computer Science*, pages 58–70. Springer, 2013. doi:10.1007/978-3-642-40328-6_5.

8 Amit Chakrabarti, Subhash Khot, and Xiaodong Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *18th IEEE Annual Conference on Computational Complexity, 2003. Proceedings.*, pages 107–117. IEEE, 2003. doi:10.1109/CCC.2003.1214414.

9 Stephen Robert Chestnut. *Stream sketches, sampling, and sabotage*. PhD thesis, Johns Hopkins University, 2015.

10 Graham Cormode, Mayur Datar, Piotr Indyk, and Shanmugavelayutham Muthukrishnan. Comparing data streams using Hamming norms (how to zero in). *IEEE Transactions on Knowledge and Data Engineering*, 15(3):529–540, 2003. doi:10.1109/TKDE.2003.1198388.

11 Graham Cormode and Donatella Firmani. A unifying framework for ℓ_0-sampling algorithms. *Distributed Parallel Databases*, 32(3):315–335, 2014. doi:10.1007/S10619-013-7131-9.

12 Rick Durrett. *Probability: theory and examples*. Cambridge University Press, 2019.

13 Philippe Flajolet, Éric Fusy, Olivier Gandouet, and Frédéric Meunier. HyperLogLog: the analysis of a near-optimal cardinality estimation algorithm. In *Proceedings of the 18th International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods for the Analysis of Algorithms (AofA)*, pages 127–146, 2007.

14 Philippe Flajolet and G. Nigel Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31(2):182–209, 1985. doi:10.1016/0022-0000(85)90041-8.

15 Sumit Ganguly. Estimating frequency moments of data streams using random linear combinations. In *Proceedings 8th International Workshop on Randomization and Computation (RANDOM)*, volume 3122 of *Lecture Notes in Computer Science*, pages 369–380, 2004. doi:10.1007/978-3-540-27821-4_33.

16 Sumit Ganguly, Mohit Bansal, and Shruti Dube. Estimating hybrid frequency moments of data streams. *Journal of Combinatorial Optimization*, 23(3):373–394, 2012. doi:10.1007/S10878-010-9339-1.

17 Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323, 2006. doi:10.1145/1147954.1147955.

18 Piotr Indyk and David Woodruff. Optimal approximations of the frequency moments of data streams. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 202–208, 2005. doi:10.1145/1060590.1060621.

19 T. S. Jayram and David P. Woodruff. The data stream space complexity of cascaded norms. In *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '09, pages 765–774, USA, 2009. IEEE Computer Society. doi:10.1109/FOCS.2009.82.

20 Daniel M. Kane, Jelani Nelson, and David P. Woodruff. An optimal algorithm for the distinct elements problem. In *Proceedings 29th ACM Symposium on Principles of Database Systems (PODS)*, pages 41–52, 2010. doi:10.1145/1807085.1807094.

21 Kevin J. Lang. Back to the future: an even more nearly optimal cardinality estimation algorithm. *CoRR*, abs/1708.06839, 2017. arXiv:1708.06839.

22 Ping Li. Estimators and tail bounds for dimension reduction in l_α $(0 < \alpha \leq 2)$ using stable random projections. In *Proceedings 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 10–19, 2008. URL: http://dl.acm.org/citation.cfm?id=1347082.1347084.

23 Yi Li, Huy L. Nguyen, and David P. Woodruff. Turnstile streaming algorithms might as well be linear sketches. In *Proceedings 46th Annual ACM Symposium on Theory of Computing (STOC)*, pages 174–183, 2014. doi:10.1145/2591796.2591812.

24 Yi Li and David P. Woodruff. Tight Bounds for Sketching the Operator Norm, Schatten Norms, and Subspace Embeddings. In Klaus Jansen, Claire Mathieu, José D. P. Rolim, and Chris Umans, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016)*, volume 60 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 39:1–39:11, Dagstuhl, Germany, 2016. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.APPROX-RANDOM.2016.39.

25 Robert Morris. Counting large number of events in small registers. *Communications of the ACM*, 21(10):840–842, 1978. doi:10.1145/359619.359627.

26 J. Ian Munro and Mike Paterson. Selection and sorting with limited storage. *Theor. Comput. Sci.*, 12:315–323, 1980. doi:10.1016/0304-3975(80)90061-4.

27 Seth Pettie and Dingyu Wang. Information theoretic limits of cardinality estimation: Fisher meets Shannon. In *Proceedings 53rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 556–569, 2021. doi:10.1145/3406325.3451032.

28 Eric Price and David P Woodruff. Applications of the Shannon-Hartley theorem to data streams and sparse recovery. In *2012 IEEE International Symposium on Information Theory Proceedings*, pages 2446–2450. IEEE, 2012. doi:10.1109/ISIT.2012.6283954.

29 Ken-Iti Sato. *Lévy processes and infinitely divisible distributions*. Cambridge University Press, 1999.

30 Dingyu Wang and Seth Pettie. Better cardinality estimators for HyperLogLog, PCSA, and beyond. In *Proceedings 42nd ACM Symposium on Principles of Database Systems (PODS)*, pages 317–327, 2023. doi:10.1145/3584372.3588680.

Randomized Lifting to Semi-Structured Communication Complexity via Linear Diversity

Vladimir Podolskii
Tufts University, Medford, MA, USA

Alexander Shekhovtsov
Moscow Institute of Physics and Technology, Russia

Abstract

We study query-to-communication lifting. The major open problem in this area is to prove a lifting theorem for gadgets of constant size. The recent paper [2] introduces semi-structured communication complexity, in which one of the players can only send parities of their input bits. They have shown that for any $m \geq 4$ deterministic decision tree complexity of a function f can be lifted to the so called semi-structured communication complexity of $f \circ \textsc{Ind}_m$, where $\textsc{Ind}_m$ is the Indexing gadget.

As our main contribution we extend these results to randomized setting. Our results also apply to a substantially larger set of gadgets. More specifically, we introduce a new complexity measure of gadgets, *linear diversity*. For all gadgets g with non-trivial linear diversity we show that randomized decision tree complexity of f lifts to randomized semi-structured communication complexity of $f \circ g$. In particular, this gives tight lifting results for Indexing gadget $\textsc{Ind}_m$, Inner Product gadget IP_m for all $m \geq 2$, and for Majority gadget MAJ_m for all $m \geq 4$. We prove the same results for deterministic case.

From our result it immediately follows that deterministic/randomized decision tree complexity lifts to deterministic/randomized parity decision tree complexity. For randomized case this is the first result of this type. For deterministic case, our result improves the bound in [6] for Inner Product gadget.

To obtain our results we introduce a new *secret sets* approach to simulation of semi-structured communication protocols by decision trees. It allows us to simulate (restricted classes of) communication protocols on truly uniform distribution of inputs.

2012 ACM Subject Classification Theory of computation → Communication complexity; Theory of computation → Oracles and decision trees

Keywords and phrases communication complexity, decision trees, lifting

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.78

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2024/199/`

1 Introduction

In recent years numerous results emerged that lift the complexity of a function in a weak model of computation to the complexity of a modified version of the function in a stronger model of computation [10, 13, 5, 11, 4, 14]. These results proved to be extremely useful for solving open problems in various areas of computational complexity [17, 18, 16, 9, 8]. In this type of results we start with a function $f\colon \{0,1\}^n \to \{0,1\}$ that is hard for a weak computation model (like decision trees) and for a gadget $g\colon \{0,1\}^m \to \{0,1\}$ we consider a function $f \circ g^n\colon \{0,1\}^{nm} \to \{0,1\}$ in which we substitute each variable of f by an output of g applied to fresh variables. Our goal is to show that the resulting function $f \circ g^n$ is hard for a strong computation model (like communication complexity or Boolean formula complexity). We would like the result to hold for as simple g, as possible.

In this paper we are mostly interested in lifting from decision tree complexity to communication complexity, which is sometimes called query-to-communication lifting. This particular type of lifting has seen numerous results [17, 10, 12, 5, 11, 4]. In these papers


© Vladimir Podolskii and Alexander Shekhovtsov;
licensed under Creative Commons License CC-BY 4.0



the results of this type were obtained and gradually improved for both deterministic and randomized cases and for gradually increasing set of possible gadgets. The size of the gadget is a parameter that is of importance for applications. The smallest known size of the gadget for both deterministic and randomized case is logarithmic. For deterministic case, the results for logarithmic size of gadgets were obtained in [5] and [20]. For randomized case, the first result was obtained in [11] with a gadget of polynomial size. The paper [4] reduced the size of gadget for randomized case to logarithm. Obtaining the lifting results with the gadgets of constant size remains a major open problem.

One of the possible approaches to this problem is to address lifting to restricted models of communication or even simpler computational models and to try to obtain lifting with constant-size gadget in this setting. Some progress in this direction was obtained in recent independent papers [2, 6].

The paper [6] shows lifting from deterministic decision tree complexity $\mathsf{D}^{\mathsf{dt}}(f)$ to deterministic parity decision tree complexity $\mathsf{D}^{\mathsf{dt}}_{\oplus}(f \circ g)$ for a wide range of gadgets g, including gadgets of constant size. More specifically, for each gadget g they introduce a stifling complexity measure k and they show that $\mathsf{D}^{\mathsf{dt}}_{\oplus}(f \circ g) \geq \Omega(k \cdot \mathsf{D}^{\mathsf{dt}}(f))$. In particular, from their result it follows that $\mathsf{D}^{\mathsf{dt}}_{\oplus}(f \circ \textsc{Ind}_m) \geq \Omega(\log m \cdot \mathsf{D}^{\mathsf{dt}}(f))$ and $\mathsf{D}^{\mathsf{dt}}_{\oplus}(f \circ \mathrm{IP}_m) \geq \Omega(\mathsf{D}^{\mathsf{dt}}(f))$ for any positive m, where $\textsc{Ind}_m$ and IP_m are Indexing and Inner Product gadgets that are among the most standard in this field (see Subsection 2.3 for the definition of these functions).

The paper [2] introduces semi-structured communication complexity. In this model one of the players is allowed to send only parities of their input bits. This model is restricted compared to regular communication complexity model, but is more powerful (up to a factor of 2) compared to parity decision trees, as players can easily simulate a parity decision tree with a semi-structured communication protocol. The paper [2] shows lifting from deterministic decision trees to semi-structured communication protocol with $\textsc{Ind}_k$ gadget for any $k \geq 4$.

Our results

We show that for a wide range of gadgets (including constant size) lifting is possible from randomized decision trees to randomized semi-structured communication complexity.

More specifically, we introduce a complexity measure *linear diversity* for gadgets. Informally, it is equal to the number of distinct (up to negation) non-constant linear functions (over $\mathbb{F}_2$) in Bob's variables we can obtain by fixing Alice's variables. We observe that the linear diversity of $\textsc{Ind}_m$ is m and the linear diversity of IP_m is $2^m - 1$.

We show that for any function (or relation) f for any $k \geq 2$ and for any gadget g with linear diversity k randomized semi-structured communication complexity of $f \circ g$ is greater or equal to $\Omega(\log k \cdot \mathsf{R}^{\mathsf{dt}}(f))$, where by $\mathsf{R}^{\mathsf{dt}}(f)$ we denote the minimal depth of probabilistic decision tree computing f. In particular, our result applies to gadgets of constant size. Our result gives tight bounds for both $\textsc{Ind}_m$ and IP_m gadgets. When lifting to probabilistic parity decision trees, our result also gives tight bounds for the MAJ_m gadget using a trick described in Subsection 3.4.

Similarly to [6] we extend our result to give the same lower bound for the logarithm of the size of the randomized semi-structured communication protocol and to a version of communication complexity, in which Bob is allowed to send indicator functions of subspaces of his input bits.

Although our techniques (see below) is designed specifically for randomized case, we translate our results to deterministic case as well. Compared to [2] the deterministic version of our results apply to a wide range of gadgets.

As an immediate corollary, we have the same results (both randomized and deterministic) for lifting to parity decision trees. Compared to [6] the deterministic version of our result for parity decision trees uses linear diversity complexity measure instead of stifling. We discuss the comparison between these two measures below.

Our results can be used to obtain lower bounds on randomized parity decision tree complexity of Boolean functions. We provide a couple of examples of bounds we can obtain.

Linear diversity vs. stifling

A function $g\colon \{0,1\}^m \to \{0,1\}$ is *k-stifled* if for any subset $S \subseteq [m]$ of k input variables and any bit b we can fix all other variables in such a way that g evaluates to b no matter what the values of the variables in the subset S are.

The binary logarithm of linear diversity measure is greater than stifling at least for some functions. A notable example is IP_m function that is a common gadget in lifting results. Its linear diversity is maximal, but its stifling is just 1.

Our techniques

The proof of our results builds on so-called simulation argument in the style of [11, 4]. In this argument, given a randomized communication protocol $\mathbf{\Pi}$ computing $f \circ g^n$ of cost d, we build a randomized decision tree $\mathbf{T}$ computing f of depth $O(d/\log k)$. The tree $\mathbf{T}$ simulates $\mathbf{\Pi}$, querying the necessary information. Next we describe the simulation argument and then explain the new ideas.

For convenience we introduce the following notations:

- The gadget: it is convenient to consider gadgets of the form $g\colon [k] \times \{0,1\}^m \to \{0,1\}$, where $[k]$ corresponds to the inputs of Alice and $\{0,1\}^m$ corresponds to the inputs of Bob, and the linear diversity of g is k; that is, for convenience, we assume that for any fixed first input of g the resulting function on the second input is linear.
- The collection of all gadgets: $G := g^n : [k]^n \times (\{0,1\}^m)^n \to \{0,1\}^n$.
- The input to the i-th gadget: $(x_i, y_i) \in [k] \times \{0,1\}^m$.
- The whole inputs of Alice and Bob: $x = (x_1, \dots, x_n)$, $y = (y_1, \dots, y_n)$.

Simulation argument. The main idea is that $\mathbf{T}$ on input $z \in \{0,1\}^n$ simulates $\mathbf{\Pi}$ on a random input (x, y) that is distributed uniformly on $G^{-1}(z)$. Since $f \circ G(x, y) = f(z)$, $\mathbf{T}(z)$ will output $f(z)$ as long as the simulation of $\mathbf{\Pi}$ performed correctly.

The main difficulty in this approach is to simulate $\mathbf{\Pi}$ on $(x, y) \sim G^{-1}(z)$ without knowing z. To overcome this problem, it was shown in [11, 4] that the distribution $(x, y) \sim G^{-1}(z)$ does not differ substantially (from the perspective of the players) from the uniform distribution on all inputs. Thus, the tree $\mathbf{T}$ can instead simulate $\mathbf{\Pi}$ on (x, y), where (x, y) is distributed uniformly on $[k]^n \times (\{0,1\}^m)^n$, until $\mathbf{\Pi}$ reveals too much information about some block of inputs (x_i, y_i). Once this happens for some i, the tree $\mathbf{T}$ queries z_i and proceeds with the simulation knowing the correct distribution of the pair (x_i, y_i).

However, in this approach the simulation becomes approximate. To be able to assume that the uniform distribution on $G^{-1}(z)$ does not differ too much from the uniform distribution on all inputs, we need that the size of the gadget is at least logarithmic in n (we need this to bound the error probability of the simulation). Thus, it is not clear how to use this approach for gadgets of smaller size.

The key idea of our approach is to simulate $\mathbf{\Pi}$ precisely on the distribution $(x, y) \sim G^{-1}(z)$. This allows us to work with the gadgets of constant size. To address the issue of the simulation with unknown z, we introduce the main key ingredient of our argument, the *secret sets* technique.

Secret sets. Let S_i be some subset of $\{i\} \times [m]$. Assume for now that z_i equals the XOR of variables of y on the positions in S_i. Then we can show that as long as S_i is not in a linear span of linear functions sent by Bob in $\mathbf{\Pi}$, the uniform distribution on $g^{-1}(z_i)$ is indistinguishable (by players) from the uniform distribution on the whole i-th block of input. This allows us to simulate $\mathbf{\Pi}$ as if z_i is known. Once S_i falls into a linear span of functions sent by Bob, we just query z_i and proceed with the simulation. To prove our bound, we must show that Bob needs to send many (about $\log_2 k$) messages in $\mathbf{\Pi}$ on average to force us to query z_i. Recall that S_i is actually random. The intuition is that Bob must send roughly $\log_2 k$ linear functions for their linear span to capture a random S_i.

There are two key ingredients to actually prove that the secret set technique forces Bob to send many messages. Below is the brief description of them.

Narrowing Bob's messages to blocks. For each Bob's message we assign a block which will account for it. For each message assigned in block $\{i\} \times [m]$, we remove the part of it that lies outside of $\{i\} \times [m]$. Denote by L_i this set of truncated messages that assigned to ith block.

Each L_i will admit the following property. S_i is not lying in the linear span of messages sent by Bob as long as it is not lying in the linear span of L_i. At some point, the property can become violated after Bob sends a message, but we will send additional messages for Alice and Bob to restore the property. More precisely, we pick a linear combination of messages assigned to ith block that annulates S_i. We subtract S_i from this linear combination, in which we take untruncated messages, and send this new message for Bob recursively.

Entropy. We use the idea of *fixed/unfixed* blocks that was also used in the previous lifting theorems that utilized entropy. At the beginning, we consider all S_i to be unfixed. Note that, initially, the binary entropy of S_i is $\log_2 k$, since x_i is uniformly distributed on $[k]$. We want to show that S_i rarely lies in the linear span of L_i when a new element is added to L_i. The case when the entropy of S_i is sufficiently larger than $|L_i|$ is favorable for us, since in this case S_i does not lie in the linear span of L_i with a high probability. When the entropy of S_i goes below that level, we fix S_i (Alice sends x_i). Since Alice and Bob must send a lot of messages to decrease the entropy of S_i below the desired level or to increase the size of L_i, the number of fixed S_i is low.

As another feature of our approach we would like to mention that unlike the previous papers our simulation algorithm has a very simple description: we fix Alice's input randomly and maintain the linear space generated by Bob's messages to decide on querying z_is. Correctness of the algorithm easily follows from its description and the most technical part of the proof shifts to the complexity analysis.

For the deterministic version of our result we again use the simulation argument in the style of [11, 4] and more specifically, our general strategy is very similar to the one of [2] (with the necessary generalization to a wide range of gadgets).

Organization

The rest of the paper is organized as follows. In Section 2 we give the necessary preliminary information and introduce key notions and notation. In Section 3 we give a formulation our results. In Subsection 3.5 we show some applications of our results. In Section 4 we introduce additional notation that is used in the proofs. In Subsection 5.1 we begin the proof of our main result and as a first step reformulate the theorem in the form that is convenient for the proof. In Subsection 5.2 we describe the protocol to simulate communication protocol by decision tree. In Subsection 5.3 we proof correctness of the simulation. In Subsection 5.4 we prove the bound on the number of queries in the simulation (this is the most technically heavy part of the proof).

Proofs of the remaining theorems are omitted due to the space constraint and are provided in the full version of the paper. The proofs of our results on lifting to the size of the communication protocols and to the subspace queries are achieved by a slight modification of the proof for the depth complexity. The proof of our results for the deterministic case are similar to the ones for randomized setting with necessary technical modifications.

2 Preliminaries

2.1 Standard Notation

Here we will describe some notation that we use. We denote $[n] := \{1, 2, ..., n\}$ for a non-negative integer n. The addition mod 2 is denoted by XOR or $\oplus$. Sometimes, we view $\{0,1\}^n$ as a vector space $\mathbb{F}_2^n$.

2.2 Computational Models

For functions of the form $f\colon \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$ we consider *semi-structured communication protocols* introduced in [2]. In these protocols Bob is only allowed to send the XOR of some subset of his input bits (and Alice is not restricted). A randomized semi-structured protocol is just a distribution over deterministic semi-structured protocols. We say that such a protocol computes the function f correctly, if on every input the probability of the correct output is at least $\frac{2}{3}$. The complexity $\mathsf{R}^{\mathsf{cc}}_{\to\oplus}(f)$ of f in this model is the minimal depth of a protocol computing f.

We also consider a subspace-query model. Consider communication protocols in which Bob is only allowed to send indicators of whether his input lies in an affine subspace of $\mathbb{F}_2^m$. We denote by $\mathsf{sR}^{\mathsf{cc}}_{\to\oplus}(f)$ the minimum complexity of a randomized protocol computing f, where the protocol is taken from the restricted class.

Additionally, let us introduce a size-complexity of a protocol. A deterministic communication protocol can be represented by a tree in which in every node either Alice or Bob sends a message. We call the size-complexity of the protocol to be the number of leafs in this tree. Denote by $\mathsf{sizeR}^{\mathsf{cc}}_{\to\oplus}(f)$ the minimum size-complexity of a randomized *semi-structured* protocol computing f.

We denote by $\mathsf{D}^{\mathsf{cc}}_{\to\oplus}(f)$, $\mathsf{sizeD}^{\mathsf{cc}}_{\to\oplus}(f)$ and $\mathsf{sD}^{\mathsf{cc}}_{\to\oplus}(f)$ the deterministic versions of these complexity measures.

We can define the semi-structured complexity measures for relations $f \subseteq (\mathcal{X} \times \{0,1\}^m) \times \mathcal{R}$ completely analogously. Here a deterministic protocol Π is said to compute f if for any $(x, y) \in \mathcal{X} \times \{0,1\}^m$ it outputs any z such that $(x, y, z) \in f$, if there is such z (the protocol can give any output if there is no such z).

For a function $f\colon \{0,1\}^n \to \{0,1\}$ we denote by $\mathsf{D}^{\mathsf{dt}}(f)$ the minimal depth of a decision tree computing f. We denote by $\mathsf{D}^{\mathsf{dt}}_{\oplus}(f)$ the minimal depth of a parity decision tree computing f (on each step such a decision tree can query an XOR of a subset of the input bits). Analogously to semi-structured communication complexity, we denote by $\mathsf{D}^{\mathsf{dt}}_{\oplus}(f)$, $\mathsf{sizeD}^{\mathsf{dt}}_{\oplus}(f)$, $\mathsf{sD}^{\mathsf{dt}}_{\oplus}(f)$, $\mathsf{R}^{\mathsf{dt}}_{\oplus}(f)$, $\mathsf{sizeR}^{\mathsf{dt}}_{\oplus}(f)$ and $\mathsf{sR}^{\mathsf{dt}}_{\oplus}(f)$ complexity measures defined by deterministic and randomized parity decision trees.

There is the following standard connection between parity decision trees and communication complexity protocols.

▶ **Lemma 1.** *For any function* $f\colon \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$ *we have*

$$\mathsf{D}^{\mathsf{cc}}_{\to\oplus}(f) \le 2\mathsf{D}^{\mathsf{dt}}_{\oplus}(f),$$

$$\mathsf{R}^{\mathsf{cc}}_{\to\oplus}(f) \le 2\mathsf{R}^{\mathsf{dt}}_{\oplus}(f).$$

Proof. Given a (randomized) parity decision tree for f Alice and Bob can use it to compute the function f by a (randomized) communication protocol. For this they simulate each query one by one computing XOR of their portion of the input and sending them to each other. Simulation of each query requires two bits of communication. ◀

2.3 Gadgets

We first define a class of gadgets that is of use for us.

▶ **Definition 2** (Family of linear functions). *The gadget $g : [k] \times \{0,1\}^m \to \{0,1\}$ is a family of linear functions of order k, if the following is true*

(1) $\forall i \in [k]$ $g(i,\cdot)$ *is a non-trivial linear function as a function of the second argument (that is, it is an XOR of a nonempty subset of its inputs),*

(2) $\forall i,j \in [k], i \neq j$ $g(i,\cdot) \neq g(j,\cdot)$.

For convenience from now on we will use the notation $g_i(y) := g(i,y)$.

We will use the following notion of gadget reduction.

▶ **Definition 3** (Gadget reduction). *Consider two gadgets $g : \mathcal{X} \times \{0,1\}^m \to \{0,1\}, h : \mathcal{Y} \times \{0,1\}^n \to \{0,1\}$. Then g is reducible to h, if there are mappings $\varphi : \mathcal{X} \to \mathcal{Y}, \psi : \{0,1\}^m \to \{0,1\}^n$, such that*

(1) $\forall x \in \mathcal{X}, y \in \{0,1\}^m$ $g(x,y) = h(\varphi(x), \psi(y))$

(2) ψ *is linear, that is* $\forall y_1, y_2 \in \{0,1\}^m$ $\psi(y_1 \oplus y_2) = \psi(y_1) \oplus \psi(y_2)$

Note that gadget reduction relation is transitive.

Gadget reduction is useful for us due to the following lemma.

▶ **Lemma 4.** *Assume that a gadget g reduces to a gadget h. Then for any relation $f \subseteq \{0,1\}^n \times \mathcal{R}$ we have*

$$\mathsf{R}^{\mathsf{cc}}_{\to\oplus}(f \circ g^n) \leq \mathsf{R}^{\mathsf{cc}}_{\to\oplus}(f \circ h^n),$$

$$\mathsf{sizeR}^{\mathsf{cc}}_{\to\oplus}(f \circ g^n) \leq \mathsf{sizeR}^{\mathsf{cc}}_{\to\oplus}(f \circ h^n).$$

$$\mathsf{sR}^{\mathsf{cc}}_{\to\oplus}(f \circ g^n) \leq \mathsf{sR}^{\mathsf{cc}}_{\to\oplus}(f \circ h^n).$$

The same inequalities are true for the deterministic complexities.

Proof. If Alice and Bob would like to compute $f \circ g^n$, they can just compute the mappings $\varphi : \mathcal{X} \to \mathcal{Y}, \psi : \{0,1\}^m \to \{0,1\}^n$ on their inputs individually and use the protocol for $f \circ h^n$. Since $\psi : \{0,1\}^m \to \{0,1\}^n$ is linear, Bob can simulate the protocol for $f \circ h^n$ sending only XORs of his input bits. To see that, denote by $e_i := \{0\}^{i-1} \times \{1\} \times \{0\}^{m-i}$. Thus, ψ is uniquely determined by $\psi(e_1), \ldots, \psi(e_m)$. Let x be Bob's input, and let $y = \psi(x)$. An arbitrary parity message of y can be represented as $\langle y, y' \rangle$ for some y'. Here, $\langle \cdot, \cdot \rangle$ is the dot product modulo 2. Observe that $\langle y, y' \rangle = \langle \psi(x), y' \rangle = \sum_i x_i \cdot \langle \psi(e_i), y' \rangle$. In other words, to compute $\langle y, y' \rangle$, it is enough to take the XOR of all those x_i for which $\langle \psi(e_i), y' \rangle$ equals one. This means that Bob can translate parity messages of y to parity messages of x. ◀

Next we define the main complexity measure for the gadgets.

▶ **Definition 5** (Linear diversity). *We let linear diversity of a function $h : \mathcal{Y} \times \{0,1\}^n \to \{0,1\}$ be the maximal k such that there is a family of linear functions $g : [k] \times \{0,1\}^m \to \{0,1\}$ of order k such that g reduces to h.*

Next we introduce a couple of standard gadgets that will be important for us.

▶ **Definition 6** (Index function). *Let* $\textsc{Ind}_m : [m] \times \{0,1\}^m \to \{0,1\}$ *be the function that on input* (i, y) *outputs* y_i.

▶ **Definition 7** (Inner product function). *Let* $IP_m : \{0,1\}^m \times \{0,1\}^m \to \{0,1\}$ *be the function that on input* (x, y) *outputs* $\bigoplus_{i=1}^{n}(x_i \wedge y_i)$.

▶ **Lemma 8.** $\textsc{Ind}_m$ *is a family of linear functions of order* m. IP_m *has linear diversity* $2^m - 1$.

Proof. The statement of the lemma is almost immediate. For $\textsc{Ind}_m$, note that for any $i \in [m]$ the output of $\textsc{Ind}_m$ is y_i, which is a linear function. For IP_m, note that for any fixed $x \in \{0,1\}^n$ the output of IP_m is $\oplus_{i:x_i=1} y_i$, which is a non-trivial linear function for each $x \neq 0$. The reduction from a family of linear functions is trivial (ψ is an identity function and φ sends an integer to its binary representation). ◀

▶ **Lemma 9.** *With probability approaching* 1 *as* m *tends to infinity, a random gadget* $g : [2^{2^m}] \times \{0,1\}^m \to \{0,1\}$ *has linear diversity at least* $2^{m/2}$. *(The values of* g *on each input are chosen randomly and independently.)*

The proof of this lemma can be found in the full version of the paper.

3 Results Statement

3.1 Randomized Semi-Structured protocols

▶ **Theorem 10.** *Let* $g : [k] \times \{0,1\}^m \to \{0,1\}$ *be a family of linear functions of order* $k \geq 2$. *Then for any relation* $f \subseteq \{0,1\}^n \times \mathcal{R}$ *we have*

$$\mathsf{R}^{\mathsf{cc}}_{\to\oplus}(f \circ g^n) \;=\; \Theta(\log_2 k \cdot \mathsf{R}^{\mathsf{dt}}(f)).$$

We note that big-O part is trivial, since Alice and Bob in communication protocol can simulate a decision tree for f and spend $O(\log_2 k)$ bits of communication for each tree query to compute the function g on the corresponding inputs.

We prove the following stronger versions of Theorem 10.

▶ **Theorem 11.** *Let* $g : [k] \times \{0,1\}^m \to \{0,1\}$ *be a family of linear functions of order* $k \geq 2$. *Then for any relation* $f \subseteq \{0,1\}^n \times \mathcal{R}$ *we have*

$$\log_2 \mathsf{sizeR}^{\mathsf{cc}}_{\to\oplus}(f \circ g^n) \;=\; \Theta(\log_2 k \cdot \mathsf{R}^{\mathsf{dt}}(f)).$$

▶ **Theorem 12.** *Let* $g : [k] \times \{0,1\}^m \to \{0,1\}$ *be a family of linear functions of order* $k \geq 2$. *Then for any relation* $f \subseteq \{0,1\}^n \times \mathcal{R}$ *we have*

$$\mathsf{sR}^{\mathsf{cc}}_{\to\oplus}(f \circ g^n) \;=\; \Theta(\log_2 k \cdot \mathsf{R}^{\mathsf{dt}}(f)).$$

For both theorems big-O part follows since $\mathsf{R}^{\mathsf{cc}}_{\to\oplus}(h) \geq \log_2 \mathsf{sizeR}^{\mathsf{cc}}_{\to\oplus}(h)$ and $\mathsf{R}^{\mathsf{cc}}_{\to\oplus}(h) \geq \mathsf{sR}^{\mathsf{cc}}_{\to\oplus}(h)$ for any function h.

As a corollary of these theorems and Lemma 4 we immediately obtain the following.

▶ **Corollary 13.** *Let* $h : \mathcal{Y} \times \{0,1\}^n \to \{0,1\}$ *have linear diversity at least* k *for* $k \geq 2$. *Then for any relation* $f \subseteq \{0,1\}^n \times \mathcal{R}$ *we have*

$$\mathsf{R}^{\mathsf{cc}}_{\to\oplus}(f \circ h^n) \;=\; \Omega(\log_2 k \cdot \mathsf{R}^{\mathsf{dt}}(f)).$$

The same result is true for $\log \mathsf{sizeR}^{\mathsf{cc}}_{\to\oplus}(f \circ h^n)$ *and* $\mathsf{sR}^{\mathsf{cc}}_{\to\oplus}(f \circ g^n)$.

3.2 Deterministic Semi-structured protocols

We translate our results to deterministic case as well.

▶ **Theorem 14.** *Let $g : [k] \times \{0,1\}^m \to \{0,1\}$ be a family of linear functions of order $k \geq 2$. Then for any relation $f \subseteq \{0,1\}^n \times \mathcal{R}$ we have*

$$\begin{aligned} \mathsf{D}^{\mathsf{cc}}(f \circ g^n) &= \Theta(\log_2 k \cdot \mathsf{D}^{\mathsf{dt}}(f)), \\ \log_2 \mathsf{sizeD}^{\mathsf{cc}}_{\to\oplus}(f \circ g^n) &= \Theta(\log_2 k \cdot \mathsf{D}^{\mathsf{dt}}(f)), \\ \mathsf{sD}^{\mathsf{cc}}_{\to\oplus}(f \circ g^n) &= \Theta(\log_2 k \cdot \mathsf{D}^{\mathsf{dt}}(f)). \end{aligned}$$

▶ **Corollary 15.** *Let $h : \mathcal{Y} \times \{0,1\}^n \to \{0,1\}$ have linear diversity at least k for $k \geq 2$. Then for any relation $f \subseteq \{0,1\}^n \times \mathcal{R}$ we have*

$$\mathsf{D}^{\mathsf{cc}}_{\to\oplus}(f \circ h^n) = \Omega(\log_2 k \cdot \mathsf{D}^{\mathsf{dt}}(f)).$$

The same result is true for $\log \mathsf{sizeD}^{\mathsf{cc}}_{\to\oplus}(f \circ h^n)$ and $\mathsf{sD}^{\mathsf{cc}}_{\to\oplus}(f \circ g^n)$.

3.3 Parity decision trees

Finally, we prove that our results imply the same results for parity decision trees instead of semi-structured communication protocols.

▶ **Theorem 16.** *Let $h : \mathcal{Y} \times \{0,1\}^n \to \{0,1\}$ have linear diversity at least k for $k \geq 2$. Then for any relation $f \subseteq \{0,1\}^n \times \mathcal{R}$ we have*

$$\mathsf{R}^{\mathsf{dt}}_{\oplus}(f \circ h^n) = \Omega(\log_2 k \cdot \mathsf{R}^{\mathsf{dt}}(f)).$$

The same is true for $\log \mathsf{sizeR}^{\mathsf{cc}}_{\to\oplus}(f \circ h^n)$ and $\mathsf{sR}^{\mathsf{cc}}_{\to\oplus}(f \circ g^n)$. The same results are also true for the deterministic complexities.

The part of this theorem concerning $\mathsf{R}^{\mathsf{dt}}_{\oplus}$ is a direct consequence of Lemma 1, Theorem 10, and Lemma 4. The same applies to $\mathsf{D}^{\mathsf{dt}}_{\oplus}$.

The part of this theorem about $\log \mathsf{sizeR}^{\mathsf{dt}}_{\oplus}$ and $\mathsf{sR}^{\mathsf{dt}}_{\oplus}$ does not follow from previous theorems that easily, and we prove them in the full version of the paper.

3.4 More powerful gadget reduction

In this section, we describe a more general version of a gadget reduction than in Definition 3. We apply the reduction to the MAJ gadget, obtaining tight bounds for lifting to parity decision trees.

More specifically, we can consider the domain $\{0,1\}^m$ of a gadget as a vector space $\mathbb{F}_2^m$ and input variables $x_1, \ldots, x_m$ as coordinates in the standard basis. The idea is that we can switch to another basis in this vector space, consider new coordinates as variables and apply the reduction in Definition 3 after that. Since the transformation to the new variables is linear, new variables can be expressed as an XOR of old variables and vice versa. Thus, the parity decision trees in new and old variables can simulate each other (this does not hold for communication complexity setting, since switching to the new basis mixes up the variables belonging to Alice and Bob).

We illustrate this idea on a MAJ gadget. Recall that MAJ_m is a function that returns 1 iff at least $m/2$ of its m variables are equal to 1.

▶ **Lemma 17.** *For any relation $f \subseteq \{0,1\}^n \times \mathcal{R}$ and $m \geq 4$, $\mathsf{R}^{\mathsf{dt}}_{\oplus}(f \circ MAJ^n_m) = \Omega(m \cdot \mathsf{R}^{\mathsf{dt}}(f))$.*

This method can be applied to gadgets other than MAJ. We formulate this approach in a theorem.

▶ **Theorem 18.** *Let $r : \{0,1\}^m \to \{0,1\}$, $h : \mathcal{X} \times \{0,1\}^l \to \{0,1\}$, $g : \mathcal{Y} \to \{0,1\}$ – be functions such that linear diversity of h is at least $k \geq 2$. Suppose that $r \circ h^m$ reduces to g after a change of basis. Then, for any relation $f \subseteq \{0,1\}^n \times \mathcal{R}$,*

$$\mathsf{R}^{\mathsf{dt}}_{\oplus}(f \circ g^n) = \Omega(\mathsf{R}^{\mathsf{dt}}(f \circ r^n) \log k)$$

By reduction we mean Definition 3. Here g depends on x, and we consider some linear change of basis $x \to y$. We also decide on some way to split variables y between Alice and Bob before applying Definition 3.

Proofs of these results are provided in the full version of the paper.

3.5 Applications

A more detailed description of applications can be found in the full version of the paper.

Recursive majority function

Let $\mathrm{MAJ}_3^{\otimes 1} := \mathrm{MAJ}_3$ be the majority function that returns 1 if at least two of its three input bits equal to 1, otherwise it returns 0.

For $k > 1$, define $\mathrm{MAJ}_3^{\otimes k}$ to be MAJ_3 in which each argument is substituted by $\mathrm{MAJ}_3^{\otimes k-1}$. Thus, $\mathrm{MAJ}_3^{\otimes k}$ takes 3^k inputs.

In [15], it is shown that

$$\Omega(2.57143^k) \leq \mathsf{R}^{\mathsf{dt}}(\mathrm{MAJ}_3^{\otimes k}) = O(2.64944^k).$$

It is easily observed that $\mathrm{MAJ}_3^{\otimes 2}$ is at least 2 linear diverse. By using it as a gadget, our lifting theorem imply that $\mathsf{R}^{\mathsf{dt}}_{\oplus}(\mathrm{MAJ}_3^{\otimes k}) = \Omega(\mathsf{R}^{\mathsf{dt}}(\mathrm{MAJ}_3^{\otimes k}))$. Thus, parity queries do not help in computing $\mathrm{MAJ}_3^{\otimes k}$ as compared to single variable queries.

The same approach can be applied to formulas having the form of complete binary AND-OR-tree. In other words, this function is obtained by repeated iteration of $f(x) = (x_1 \vee x_2) \wedge (x_3 \vee x_4)$. It was shown in [19] that R^{dt} of this function is at least $n^{0.7537\ldots}$, where n is the number of inputs. Since f can be used as a gadget, we translate this result to parity decision trees.

Quantum Complexity

Our lifting theorem can be applied to exhibit a separation between randomized parity decision tree complexity and bounded-error quantum complexity. Let $k \leq \log n$. It was shown in [1] that there is a $\lceil k/2 \rceil$ versus $\widetilde{\Omega}(n^{1-\frac{1}{k}})$ separation between the quantum and randomized query complexity. The separation was shown for a partial function f called k-fold Forrelation.

Using our result, we can lift this separation to randomized parity query complexity. Consider the function $f \circ \textsc{Ind}_2^n$. On the one hand, its quantum complexity is the same as the quantum complexity of f (up to a factor), since a quantum protocol can extract inputs to f from $\textsc{Ind}_2$ in constant number of additional queries. On the other hand, by Theorem 10, the randomized parity query complexity of $f \circ \textsc{Ind}_2^n$ is no less than its randomized query complexity. Hence, we obtain the same separation even if our query model can ask parity queries.

An alternative method to obtain separation between randomized parity query and quantum complexity was given in [3]. They also relied on the result of [1]. They used Fourier analysis and properties of k-Fold Forrelation to derive their result. In contrast, our lifting theorem can lift any function f that exhibits the separation.

4 Notation for proving main theorems

In this section we introduce notation and facts about entropy that will be used for proving the main theorems in the subsequent sections. Recall that in a semi-structured communication protocol, Bob can send only parities of its input. In a lifting setting, Bob is given $n \cdot m$ boolean variables. For each of n variables of the initial function, there are m gadget's variables. To analyze Bob's messages, we introduce the following definitions.

4.1 Notation

▶ **Definition 19** (XOR of a subset). *For a set $S \subseteq [m]$ and $y \in \{0,1\}^m$, we define*

$$y_S := \bigoplus_{j \in S} y_j.$$

Analogously, we define y_S if $y \in (\{0,1\}^m)^n$ and $S \subseteq [n] \times [m]$.

If $y \in (\{0,1\}^m)^n$ is the Bob's input, then each of his messages can be represented as y_S for some $S \subseteq [n] \times [m]$.

▶ **Definition 20** (Subsets of $[n] \times [m]$ as a linear space). *We view subsets of $[n] \times [m]$ as vectors in a linear space over $\mathbb{F}_2$ (where addition (+) corresponds to symmetric difference).*

With this notation, for any $S, T \subseteq [n] \times [m], y \in (\{0,1\}^m)^n$, $y_{S+T} = y_S \oplus y_T$.

▶ **Definition 21** (Linear Order on $[n] \times [m]$). *We introduce a lexicographic linear order on $[n] \times [m]$. A pair $(i,j) \in [n] \times [m]$ is said to be lower than (i',j') if $i < i'$ or $i = i', j < j'$.*

▶ **Definition 22** (Principal variable of a non-empty subset). *For a non-empty set $S \subseteq [n] \times [m]$, denote by $p(S)$ its lowest element.*

▶ **Definition 23** (Block). *We refer to $\{i\} \times [m]$ as i-th block. A non-empty subset $S \subseteq [n] \times [m]$ touches i-th block if $p(S) \in \{i\} \times [m]$.*

▶ **Definition 24** (Bernoulli distribution). *Denote by $Bern(p)$ a distribution of a random variable that takes value 0 with probability $1-p$ and 1 with probability p.*

▶ **Definition 25** (Entropy). *For a random variable x, denote by $H(x)$ its binary entropy. If X is a set, then we define its entropy as $H(X) := \log_2 |X|$. If p is a number, then $H(p)$ denotes the binary entropy of the distribution $Bern(p)$.*

Recall that the binary entropy of a random variable taking n values with non-zero probabilities $p_1, ..., p_n$ equals to

$$\sum_{i=1}^{n} -p_i \log_2 p_i$$

4.2 Entropy theorems

We state well-known results that will be used for proving our main theorems.

▶ **Lemma 26** (Gibbs' inequality [7]). *Let $0 \le p \le 1, 0 < q < 1$. Then,*

$$H(p) \le p \log_2 \frac{1}{q} + (1-p) \log_2 \frac{1}{1-q}$$

▶ **Lemma 27** (Fano's Inequality [7]). *Let X, Y be random variables. Let $\varepsilon = P(X \neq Y) \le 1/2$. Then,*

$$H(X|Y) \le H(\varepsilon) + \varepsilon \log_2(|\mathcal{X}| - 1),$$

where $\mathcal{X}$ denotes the support of X.

5 Proof of Theorem 10

5.1 Reformulation of Theorem 10

As we noted in Section 3, the $\le$-direction is simple. Thus, it remains to prove the other direction.

Let $\mathbf{\Pi}$ be a randomized semi-structured protocol computing $f \circ g^n$. Denote by d the depth of $\mathbf{\Pi}$. Our goal is to construct a randomized tree $\mathbf{T}$ of depth $O(d/\log_2 k)$, that computes $f(z)$ for any given z.

The idea is to simulate $\mathbf{\Pi}$ on a randomly uniformly chosen pair (x, y) satisfying $g^n(x, y) = z$. For convenience, denote $P_z = \{(x, y) \mid g^n(x, y) = z\}$. For any pair $(x, y) \in P_z$ we have that $\mathbf{\Pi}(x, y) = f(z)$ with probability at least $2/3$. Thus, if we sample $(x, y) \sim U(P_z)$, the protocol $\mathbf{\Pi}(x, y)$ will also be equal to $f(z)$ with probability at least $2/3$.

We use the following general strategy for $\mathbf{T}$: sample $(x, y) \sim U(P_z)$, sample $\Pi \sim \mathbf{\Pi}$ (recall that $\mathbf{\Pi}$ is a random distribution on deterministic protocols), and output $\Pi(x, y)$. Note that the choice of the pair (x, y) is independent from the choice of Π, and thus it does not matter in which order we sample Π and (x, y). As a result, what is left to do is to simulate the given deterministic semi-structured protocol Π on a random pair $(x, y) \sim U(P_z)$.

We are going to prove the following intermediate theorem.

▶ **Theorem 28.** *Let Π be a deterministic semi-structured protocol with inputs from $[k]^n \times (\{0,1\}^m)^n$. Let d be equal to the depth of Π. Then, there exists a randomized decision tree $\mathbf{T}$ which, on input $z \in \{0,1\}^n$, outputs a random variable that is destributed as $\Pi(x, y)$ for $(x, y) \sim U(P_z)$. The expected number of queries to z made by $\mathbf{T}$ is $O(d/\log_2 k)$, where the expectation is taken over (x, y) for a fixed z.*

Before proceeding to the proof of Theorem 28 we show how to prove Theorem 10 based on Theorem 28.

Proof of Theorem 10. The computation of f on input z proceeds as follows. We choose a random $\Pi \sim \mathbf{\Pi}$ and run $\mathbf{T}$ provided by Theorem 28 on input z. By definition, $\mathbf{T}(z)$ has the same distribution as $\mathbf{\Pi}(x, y)$ for $(x, y) \sim U(P_z)$. Thus, $\mathbf{T}$ computes $f(z)$ with probability at least $\frac{2}{3}$.

The average number of queries made by $\mathbf{T}$ is at most $O(d/\log_2 k)$. To achieve this number of queries in the worst case, we halt $\mathbf{T}$ if it makes 10 times more queries than the expected number. By Markov's inequality, this only happens with probability at most $\frac{1}{10}$, the probability of the correct answer is still a constant greater than $1/2$ and it can be increased to $2/3$ by the standard argument for error reduction. ◀

Now we proceed to the proof of Theorem 28. For this we need to describe how $\mathbf{T}$ simulates Π. In the subsequent sections we will heavily use the notation introduced in Section 4.

5.2 Simulation algorithm for T

In this section we describe the algorithm for $\mathbf{T}$.

$\mathbf{T}$ starts by sampling a random $a \sim U([k]^n)$ and assuming that $x = a$.

After that, $\mathbf{T}$ starts the simulation of Π. Since x is fixed ($\mathbf{T}$ knows x but for Π it is a random variable), Alice's messages are easily simulated. Next, we describe how $\mathbf{T}$ simulates Bob's parity messages on the (unknown) variables y.

Recall that we can view g as a family of linear functions $\{g_1, \dots, g_k\}$ of order k (one function for each value of x). Since $g_1, ..., g_k$ are linear functions, we can represent them as $g_j(x) = x_{G_j}$ for some $G_j \subseteq [m]$.

Let $S_i := \{i\} \times G_{x_i}$. Note that all S_is are linearly independent since they are in distinct blocks. From now on we will call $S_1, ..., S_n$ *the secret sets*.

Consider an arbitrary step of simulation and assume Bob has already sent the parities of his inputs for the sets $Q_1, ..., Q_t \subseteq [n] \times [m]$ and is now supposed to send the parity of y's in the set $Q_{t+1} \subseteq [n] \times [m]$. Note, that it can be assumed that Q_{t+1} is linearly independent of $Q_1, ..., Q_t$: otherwise, the message Q_{t+1} does not reveal any new information and can be omitted from the protocol. There are two cases:

(1) There is a linear combination of Q_is that includes Q_{t+1} and equals to some linear combination of the secret sets:

$$Q_{i_1} + Q_{i_2} + ... + Q_{i_k} + Q_{t+1} = S_{j_1} + ... + S_{j_l}.$$

In this case, the value $y_{Q_{t+1}}$ is uniquely determined since

$$y_{Q_{t+1}} = y_{Q_{i_1}} + y_{Q_{i_2}} + ... + y_{Q_{i_k}} + z_{j_1} + ... + z_{j_l}.$$

The y's parities on the sets $Q_{i_1}, ..., Q_{i_k}$ are already known from the previous messages of Bob. Therefore, $\mathbf{T}$ queries $z_{j_1}, ..., z_{j_l}$ (if it hasn't already), and we calculate the value $y_{Q_{t+1}}$ that Bob sends in this vertex.

(2) Such a linear combination does not exist. In this case, $\mathbf{T}$ sends a random bit $Bern(1/2)$ as an XOR of y on the set Q_{t+1}. In terms of the protocol Π, this corresponds to proceeding to one of the left and right children with equal probabilities.

Thus, we have described how to simulate each Bob's message. Once $\mathbf{T}$ reaches a leaf of Π, it outputs the value written in this leaf.

5.3 Correctness of the simulation of Π done by T

Next, we need to show that $\mathbf{T}(z)$ indeed simulates Π on a random pair $(x, y) \sim U(P_z)$, and that $\mathbf{T}(z)$ makes $O(d/\log_2 k)$ queries on average. We start with the correctness of the simulation.

It is easy to see that for any z the projection of $U(P_z)$ onto x is the uniform distribution on $U([k]^n)$, therefore fixing $x = a \sim U([k]^n)$ correctly corresponds to the distribution of (x, y) we would like to simulate the protocol on.

Note that once x is fixed the only constraints on y are of the form $g_j(y_i) = z_i$, where $j = x_i$. In particular, any variable in y, not included in $\{i\} \times G_{x_i}$, has a $Bern(1/2)$ distribution.

Next, we justify why the algorithm can send $Bern(1/2)$ as an XOR of Q_{t+1} if no linear combination exists (Case 2 above). Indeed, the y's parities of $Q_1, ..., Q_t, S_1, ..., S_n$ define an affine subspace in our linear space. Since Q_{t+1} is linearly independent of $Q_1, ..., Q_t, S_1, ..., S_n$, each of the conditions $y_{Q_{t+1}} = 0$ and $y_{Q_{t+1}} = 1$ is satisfied by exactly half of the points of the affine subspace.

If there exists a linear combination of Q_is that includes Q_{t+1} and equals a linear combination of S_is (Case 1 above), then given the previous messages and the value of z there is only one possible value for $y_{Q_{t+1}}$, which $\mathbf{T}$ indeed sends in our simulation.

Thus, the distribution of $\mathbf{T}(z)$ matches $\Pi(x, y)$ for $(x, y) \sim U(P_z)$.

5.4 Upper bound on the average number of queries made by T

Next we show that $\mathbf{T}(z)$ makes $O(d/\log_2 k)$ queries to the variables z on average. For this we introduce some complexity measures and study their behaviour during the execution of the protocol. To make this analysis cleaner we first modify the protocol Π to add more messages to it. This does not change the output of the protocol, but it allows us to simplify the exposition of time analysis. In the next section we describe the modified protocol. Next we discuss its connection to $\mathbf{T}$. Finally, in Subsubsection 5.4.3, we derive the upper bound on the number of queries made by $\mathbf{T}$.

Since we need to show the upper bound on the number of queries of $\mathbf{T}(z)$ for any z, we fix z for the whole argument.

5.4.1 Description of the modified protocol $\overline{\Pi}_z$

Here we describe $\overline{\Pi}_z$, a refined version of Π. Note that the protocol depends on z, which is fixed throughout the whole argument. We will be interested only in the behaviour of the protocol on inputs in P_z.

We modify the protocol in two ways. First, we modify the messages sent by Bob into equivalent messages. This does not actually change the information transmitted by Bob, this modification is needed only for the purpose of the analysis of the protocol. Next, at some moments of time we add additional messages sent by the players. The information transmitted in these messages is not affecting the following messages in the protocol. One can think of this in the following way: at some points of the protocol the players pause the execution of the protocol, exchange some extra information, and then resume the execution of the protocol as if nothing happened. These extra messages are needed just to introduce new intermediate vertices in communication tree that will allow us to analyse the complexity of $\mathbf{T}$ in a cleaner way.

The pseudocode for the algorithm $\overline{\Pi}_z$ is provided in Figure 1. Next we describe the protocol and introduce some important notation related to it.

First of all, observe that if Bob sends b_1, b_2 as parities on the sets $Q_1, Q_2 \subseteq [n] \times [m]$ respectively, this is equivalent to sending b_1, $b_1 \oplus b_2$ as parities on $Q_1, Q_1 + Q_2$ respectively. More generally, applying an invertible linear transformation to Bob's messages does not change the information transmitted.

Recall that by $S_i = \{i\} \times G_{x_i}$ we denote the secret sets. Note that S_i depends on x and, thus, initially is only known to Alice. Initially, all S_i are *unrevealed*, and over the course of the simulation, they will be gradually *revealed*. We also introduce a separate notion that of S_i being *fixed*. Initially, all S_i are *unfixed* and then will change status to fixed over the course of the simulation. A revealed S_i will also be fixed, but not necessarily vice versa.

Suppose that at the current moment of simulating Π, Bob has sent parities on the sets $Q_1, ..., Q_t \subseteq [n] \times [m]$ and now he wants to send the parity on the set $Q \subseteq [n] \times [m]$. We will maintain the *principle variable invariant*: all $p(Q_i)$ are distinct and, if $i < j$, then $p(Q_i)$ is *lower* than $p(Q_j)$. The variables $p(Q_i)$ will be referred to as *principal*. In particular, from this invariant, it follows that $Q_1, ..., Q_t$ are linearly independent.

Define the procedure *sift*, which will transform Q to an equivalent message. The procedure *sift* iterates over $i = 1, \dots, t$ and replaces Q with $Q + Q_i$ if $p(Q_i) \in Q$. Note that after this procedure for any i we have that $p(Q_i) \notin Q$.

We run the procedure *sift* on Q. If it turns into an empty set, then Bob's message does not provide any new information. In this case we finish its processing and move on to further simulation of Π. If the resulting Q is not empty, then it contains a principal variable. Since after *sift*, Q does not contain principal variables of previous messages, the principal variable of Q does not coincide with the principal variable of previous messages. We insert a copy of Q into the sequence $Q_1, ..., Q_t$ in such a way that the principle variable invariant is maintained. That is, now the length of the sequence is $t + 1$. Assume that $p(Q)$ is in the i-th input block. Let $L_i := \{Q_j \cap \{i\} \times [m] \mid p(Q_j) \in \{i\} \times [m]\}$. In other words, we consider all sets whose principal variables are in the i-th block and intersect them with the i-th block. Note that sets in L_i are linearly independent. Denote by $\mathcal{L}_i$ the linear span of L_i with the zero vector (empty set) removed.

At this point, we apply the second modification to the protocol. If the binary entropy of S_i is less than $\frac{1}{10}\log_2 k + \log_2 |\mathcal{L}_i| + \frac{1}{3}$, Alice sends S_i, and S_i is considered fixed. Here the protocol views S_i to be a random variable of the distribution $(x, y) \sim U(P_z)$ conditioned on the information about (x, y) that the protocol has learnt so far. Note that we fixed z in advance, and we assume that the inputs given to the players are indeed in P_z (we are not interested in the behaviour of the protocol on other inputs). As a result, both players can compute the entropy of S_i and compare it to $\frac{1}{10}\log_2 k + \log_2 |\mathcal{L}_i| + \frac{1}{3}$ without communication.

After that Alice sends a message indicating if S_i lies in $\mathcal{L}_i$[1]. If this is not the case or if S_i has already been revealed on one of the previous steps, we finish processing the message Q and proceed with the further simulation of the protocol Π. If S_i lies in $\mathcal{L}_i$, then we consider S_i to be revealed. In this case Alice sends S_i, and we fix S_i if it was not already fixed before. Bob sends y_{S_i}. Next, let $Q_{j_1}, ..., Q_{j_l}$ be the sets whose linear combination, when intersected with the i-th block, equals the secret set S_i. The set Q must be present among them: otherwise, S_i would have been revealed at an earlier step. Without loss of generality we can assume that $Q_{j_1} = Q$. We update Q to $Q \leftarrow Q + Q_{j_2} + ... + Q_{j_l} + S_i$ and perform the same procedure with the updated Q starting with *sift* (note that from this players can compute y_Q for the new Q without additional communication since y_{S_i} is known). Note, that during this iteration we removed from Q all elements in the i-th block without introducing anything to Q in the previous blocks (all sets in the combinations had their principle variables in the i-th block). As a result, the updated Q lies within the blocks that are higher than i-th block and the whole procedure of updating Q will be finished eventually.

5.4.2 Connection between T and $\overline{\Pi}_z$

The protocol $\overline{\Pi}_z$ is useful for upper bounding the complexity of $\mathbf{T}$ due to the following lemmas.

[1] Note that this might be redundant if we just communicated the whole S_i. We choose to keep this step even if it is redundant to make the pseudocode in Figure 1. In the analysis below the redundancy of these messages is reflected in Observation 35.

Refined protocol $\overline{\Pi}_z$ ***on input*** $(x, y) \sim U(P_z)$**:**

1: Initialize: v = root of Π, $Q_1, ..., Q_t \subseteq [n] \times [m]$ – sets which parities Bob sends, initially $t = 0$
2: **while** v is not a leaf [invariant: $p(Q_i)$ is lower than $p(Q_j)$ when $i < j$]
3: Let v_0, v_1 be children of v
4: **if** Bob speaks at v **then**
5: Let $Q \subseteq [n] \times [m]$ be the set which parity Bob sends at v
6: Let $b = y_Q$
7: ▷ Bob sends b and we update $v \leftarrow v_b$ ▷ (B1)

8: **for** $i = 1..t$ **do**
9: **if** $p(Q_i) \in Q$ **then**
10: $Q \leftarrow Q + Q_i$
11: **end if**
12: **end for**
13: **if** $Q \neq \varnothing$ **then**
14: Insert a copy of Q in $Q_1, ..., Q_t$ so that invariant holds
15: Let $i \in [n]$ be the block containing $p(Q)$
16: // $L_i = \{Q_j \cap \{i\} \times [m] \mid p(Q_j) \in \{i\} \times [m]\}$
17: // $\mathcal{L}_i$ denotes linear span of L_i without null vector
18: // $S_i = \{i\} \times G_{x_i}$ – a *secret* set in ith block
19: **if** $H(S_i) < \frac{1}{10} \log_2 k + \log_2 |\mathcal{L}_i| + \frac{1}{3}$ **then**
20: ▷ Alice sends S_i; S_i is *fixed* now ▷ (A3)
21: **end if**
22: ▷ Alice communicates whether S_i is in $\mathcal{L}_i$ ▷ (A2)
23: **if** $S_i \in \mathcal{L}_i$ and S_i is not *revealed* **then**
24: ▷ Alice sends S_i; S_i is *fixed* now ▷ (A3)
25: S_i is considered to be *revealed*
26: ▷ Bob sends y_{S_i} ▷ (B2)
27: Find distinct $j_1, ..., j_l$, s.t. $Q = Q_{j_1}$ and $(Q_{j_1} + ... + Q_{j_l}) \cap \{i\} \times [m] = S_i$
28: $Q \leftarrow Q + Q_{j_2} + ... + Q_{j_l} + S_i$
29: Go to 8−th line
30: **end if**
31: **end if**

32: **else** Alice speaks at v
33: Let b be the bit she sends
34: ▷ Alice sends b and we update $v \leftarrow v_b$ ▷ (A1)
35: **end if**
36: **end while**
37: return the value of the leaf v

Figure 1 The modified (deterministic) protocol $\overline{\Pi}_z$. The original protocol Π can be recovered by ignoring lines 8–31 and the **red** text. Lines 8–31 are used to maintain the invariant and prove the estimate on the number of *revealed* sets. The classification (A1), (A2), (A3), (B1), (B2) of the actions made by Alice and Bob is used in Section 5.4.3. Note that Alice can send more than 1-bit of information for the message of type (A3).

▶ **Lemma 29.** *The following is a while-loop invariant in $\overline{\Pi}_z$: If a linear combination of $Q_1, ..., Q_t$ equals a linear combination of $S_1, ..., S_n$, then all S_is in this linear combination are revealed.*

Proof. First note that for all i such that S_i is unrevealed, S_i does not lie in $\mathcal{L}_i$. Otherwise, at the line 25 of the algorithm, S_i would have been revealed.

Assume that this invariant is violated. Let this linear combination be $Q_{j_1} + ... + Q_{j_l} = I$, where $j_1 < ... < j_l$. If there are many linear combinations that violate invariant, then choose the one with the greatest j_1. Let i be the block of the variable $p(Q_{j_1})$. Then, $I \cap \{i\} \times [m]$ must be equal to S_i, since $I \cap \{i\} \times [m] \neq \varnothing$. It follows that S_i must be revealed, since it lies in $\mathcal{L}_i$. At the line 28 of the algorithm, it is ensured that a linear combination $Q_{j_1} + ... + Q_{j_l} + S_i = I + S_i$ is added to the set of Qs. $I + S_i$ is contained in blocks strictly higher than i-th block and, by assumption, is equal to a linear combination of $S_1, ..., S_n$ containing an unrevealed secret set. Therefore, we obtained a contradiction with the maximality of j_1. ◀

Next we establish the connection between $\mathbf{T}$ and $\overline{\Pi}_z$. By construction, $\mathbf{T}(z)$ simulates Π on a random input $(x, y) \sim U(P_z)$. Fix r to be the outcome of the random bits of the tree $\mathbf{T}$. Then $\mathbf{T}_r(z)$ will simulate Π on some pair $(x_r, y_r) \in P_z$.

We show the following.

▶ **Lemma 30.** *The number of queries to z made by $\mathbf{T}_r(z)$ is less or equal than the number of revealed sets in protocol $\overline{\Pi}_z$ on input (x_r, y_r).*

Proof. By definition of $\mathbf{T}$, it queries z_i only if there is some linear combination of $S_1, ..., S_n$ containing S_i that equals a linear combination of $Q_1, ..., Q_t$. By Lemma 29, we get that z_i is queried by $\mathbf{T}$ only if S_i is revealed. Hence, the statement of the lemma follows. ◀

The lemma holds for all r. In particular, if we average over r, then we get that the expected number of queries made by $\mathbf{T}$ is bounded by the expected number of revealed sets in protocol $\overline{\Pi}_z$ on a random $(x, y) \sim U(P_z)$. Therefore, it remains to obtain an upper bound on the expected number of revealed sets.

5.4.3 Upper bound on the expected number of revealed sets in $\overline{\Pi}_z$

For each vertex v of the protocol $\overline{\Pi}_z$, define

$$P_{z,v} = \{(x, y) \mid \overline{\Pi}_z \text{ reaches vertex } v \text{ on input } (x, y) \text{ and } g^n(x, y) = z\}$$

In other words, if $X_v \times Y_v$ is the rectangle corresponding to the vertex v in the protocol $\overline{\Pi}_z$, then $P_{z,v} = X_v \times Y_v \cap (g^n)^{-1}(z)$.

Essentially, if the protocol $\overline{\Pi}_z$ has reached the vertex v, then (x, y) can be any element of the set $P_{z,v}$. Now consider the uniform distribution $U(P_z)$ and run $\overline{\Pi}_z$ on a random pair (x, y) from this distribution. Then $U(P_{z,v})$ is precisely the conditional distribution of the pair (x, y), given that $\overline{\Pi}_z$ has reached v. In what follows, we consider $(x, y) \sim U(P_{z,v})$. We will be interested in the entropy of the projection of this distribution onto x, hence we introduce the following notation

$$H^v = H(x),$$

$$H^v_i = H(x_i).$$

Let S_i be the secret set in the i-th block. The set S_i depends on x_i and, moreover, there is a one-to-one correspondence between S_is and x_is. Therefore, their entropies are equal: $H(S_i) = H(x_i) = H_i^v$. Denote by L_i^v the set L_i corresponding to the vertex v, and by $\mathcal{L}_i^v$ its linear span with the zero vector removed. That is, $|\mathcal{L}_i^v| = 2^{|L_i^v|} - 1$.

Using Fano's inequality, we can show the following.

▶ **Lemma 31.** *Let v be a vertex of the protocol $\overline{\Pi}_z$ in which Alice sends a message revealing whether S_i lies in the linear span L_i^v (this is (A2) type of message on Figure 1). Then, if $H_i^v \geq \log_2 |\mathcal{L}_i^v| + \frac{1}{10}\log_2 k + \frac{1}{3}$, then S_i does not lie in $\mathcal{L}_i^v$ with probability at least $\frac{1}{100}$.*

Proof. Define Y to be equal to S_i if $S_i \in \mathcal{L}_i^v$, and to be equal to any element in $\mathcal{L}_i^v$ otherwise. Note that with this definition, $P(S_i \notin \mathcal{L}_i^v) = P(S_i \neq Y)$. Denote this probability by ε. Then, by Fano's inequality we have

$$H(S_i) - \log_2 |\mathcal{L}_i^v| \leq H(S_i) - H(Y) \leq H(S_i|Y) \leq H(\varepsilon) + \varepsilon \log_2 k.$$

Assume that $\varepsilon < 1/100$. Then

$$H(S_i) < \frac{1}{10}\log_2 k + \log_2 |\mathcal{L}_i^v| + H(1/100).$$

Since $H(1/100) < 1/3$, we get at a contradiction with the statement of the lemma. ◀

Now let's show that if at vertex v Alice or Bob sends one-bit message, then the entropy $H(x)$ on average does not decrease significantly.

▶ **Lemma 32.** *Let $I_v : P_{z,v} \to \{0,1\}$ denote the bit of information that Alice or Bob sends at vertex v. Then*

$$H(x|I_v) \geq H(x) - H(I_v) \geq H(x) - 1.$$

Proof. We have that

$$H(x|I_v) + H(I_v) = H(x, I_v) \geq H(x)$$

and

$$H(x|I_v) \geq H(x) - H(I_v) \geq H(x) - 1.$$ ◀

In other words, the entropy H of the distribution of x drops by no more than 1 on average on one step of the protocol. Next, we introduce the deficiency of the distribution. Let

$$U_i^v = \begin{cases} \log_2 k, & \text{if } S_i \text{ is not fixed;} \\ 0, & \text{if } S_i \text{ is fixed,} \end{cases}$$

and let the deficiency be

$$D^v = \left(\sum_{i=1}^n U_i^v\right) - H^v.$$

Note that $H(S_i) = 0$ if S_i is fixed. Thus, $D^v \geq 0$ for any v, as $\sum_{i=1}^n U_i^v \geq \sum_{i=1}^n H_i^v \geq H^v$. We will omit the superscript v if the vertex is clear from the context.

Now we consider $(x, y) \sim U(P_z)$ and introduce some random variables for various types of messages of $\overline{\Pi}_z$ on the input (x, y).

For this we consider a finer classification of the messages sent by Alice, compared to the one in Figure 1:

(A1) A message that Alice sends in Π.

(A2') A message $S_i \overset{?}{\in} \mathcal{L}_i$, such that we have $H_i \geq \log_2 |\mathcal{L}_i| + 1/3 + \frac{1}{10}\log_2 k$ before sending this message. In other words, in these messages S_i was not previously fixed.

(A2") A message $S_i \overset{?}{\in} \mathcal{L}_i$, such that we have $H_i < \log_2 |\mathcal{L}_i| + 1/3 + \frac{1}{10}\log_2 k$ before sending this message. In other words, in these messages S_i was previously fixed.

(A3') A message that Alice sends to find out the exact value of S_i, and S_i was not fixed before sending this message.

(A3") Same as (A3') but S_i was fixed before sending this message.

For Bob we use the same classification of messages as in Figure 1:

(B1) A message that Bob sends in Π.

(B2) A message in which Bob communicates the value y_{S_i}

Let $A_1, A'_2, A''_2, A'_3, A''_3, B_1, B_2$ denote the number of messages of the corresponding type sent by the protocol during the whole computation. All of these are random variables depending on (x, y), which we draw randomly: $(x, y) \sim U(P_z)$.

The following inequality hold.

▶ **Observation 33.** $\mathbb{E}A'_2 \leq 100 \cdot \mathbb{E}B_1$.

Proof. When Alice sends a message of type (A2'), by Lemma 31 the secret set S_i does not lie in $\mathcal{L}_i$ with probability at least $\frac{1}{100}$. If it does not lie in $\mathcal{L}_i$, the processing of the Bob's message is over. Thus, for each query of type (B1), there will be no more than 100 queries of type (A2') on average. ◀

▶ **Observation 34.** *At the end of the computation, the number of fixed S_i is greater or equal than A'_3. In particular, there are at least A'_3 coordinates i such that U_i decreased from $\log_2 k$ to 0. During each of these decreases of U_is, D decreases by at least $\max\{0, \frac{9}{10}\log_2 k - 1/3 - \log_2 |\mathcal{L}_i|\}$ on average.*

Proof. By definition, A'_3 is precisely equal to the number of the fixed secret sets S_i. When the status of S_i changes from unfixed to fixed, U_i drops from $\log_2 k$ to 0. Since U_i changes to 0 only when H_i becomes less than $\log_2 |\mathcal{L}_i| + 1/3 + \frac{1}{10}\log_2 k$ (as ensured by if's on lines 19 and 23), sending S_i transmits no more than $\log_2 |\mathcal{L}_i| + 1/3 + \frac{1}{10}\log_2 k$ information. Thus, we get the required decrease of D by at least $\max\{0, \frac{9}{10}\log_2 k - 1/3 - \log_2 |\mathcal{L}_i|\}$ on average. ◀

▶ **Observation 35.** *Messages of types (B2), (A2"), and (A3") do not affect D.*

Proof. A message of the type (B2) does not change D since pairs $(x, y) \in P_z$ have a property that $g(x_i, y_i) = z_i$. Thus, message y_{S_i} does not decrease the entropy of the distribution.

Just before a message M of type (A2") is sent, by definition, we have that $H_i < \log_2 |\mathcal{L}_i| + 1/3 + \frac{1}{10}\log_2 k$. But in lines $19 - 21$ it is ensured that if H_i is lower than this threshold, then S_i will be fixed. Thus, H_i is not only lower than $\log_2 |\mathcal{L}_i| + 1/3 + \frac{1}{10}\log_2 k$, but also equals to 0. Thus, $(A2'')$ does not influence D.

By definition, S_i is fixed before sending a message of type (A3"). Thus, $H_i = 0$ and sending S_i does not reveal any information. ◀

▶ **Observation 36.** *At the end of the protocol's execution, $\sum_i |L_i| \leq B_1 + B_2$.*

Proof. A new set Q is generated and added to the list of Q_is either when a message of type (B1) is sent, or after a message of type (B2) is sent. ◀

Let's show how the desired bound follows from these lemmas. Note that if S_i is revealed, then it must be fixed. As a consequence, we get $B_2 \le A'_3$ since B_2 equals the number of revealed sets and A'_3 equals the number of fixed ones. Our goal is to upper bound the number of revealed sets. Thus, we only need to upper bound $\mathbb{E}A'_3$ by $\mathbb{E}(A_1 + B_1)/\log_2 k$, as $A_1 + B_1$ is exactly the number of messages sent by the protocol Π.

▶ **Lemma 37.** $\mathbb{E}A'_3 = O(\mathbb{E}(A_1 + B_1)/\log_2 k)$

Proof. By Lemma 32 messages of type (A1) and (B1) increase D by no more than 1 on average. The same is true for messages of type (A2'). As a result, due to Observation 34 and Observation 35,

$$\mathbb{E}\left(-A'_3 \cdot \left(\frac{9}{10}\log_2 k - 1/3\right) + \sum_i \log_2 |\mathcal{L}_i| + A_1 + B_1 + A'_2\right) \ge 0,$$

since D is always greater or equal than than 0. Applying Observation 33 and rearranging we get

$$\mathbb{E}\left(\sum_i \log_2 |\mathcal{L}_i| + 101 \cdot B_1 + A_1\right) \ge \mathbb{E}A'_3 \cdot \left(\frac{9}{10}\log_2 k - 1/3\right).$$

Now we consider two cases.

1. $k \ge 3$
   Note that $\sum_i \log_2 |\mathcal{L}_i| \le \sum_i |L_i| \le B_1 + B_2$. Using the fact that $B_2 \le A'_3$ and the inequality above, we obtain

   $$\mathbb{E}\left(102 \cdot B_1 + A_1\right) \ge \mathbb{E}A'_3 \cdot \left(\frac{9}{10}\log_2 k - 4/3\right).$$

   Since $\frac{9}{10} \cdot \log_2 k > 4/3$, we get $\mathbb{E}A'_3 = O\left(\frac{\mathbb{E}A_1 + B_1}{\log_2 k}\right)$.
2. $k = 2$
   Note that if we are sending a message of type (A3') and it is true that $|L_i| \le 1$, then D decreases by at least $1 - (1/10 + 1/3) \ge \frac{1}{2}$ on average. The number of is such that $|L_i| \ge 2$ does not exceed $\frac{B_1+B_2}{2}$ by Observation 36. As a result,

   $$\mathbb{E}\left(-\frac{1}{2} \cdot \left(A'_3 - \frac{B_1 + B_2}{2}\right) + A_1 + B_1 + A'_2\right) \ge 0.$$

   Using $B_2 \le A'_3$, we get

   $$\mathbb{E}\left(5/4 \cdot B_1 + 100 \cdot B_1 + A_1\right) \ge \mathbb{E}A'_3 \cdot 1/4.$$

   Thus, we again obtain $\mathbb{E}A'_3 = O(\mathbb{E}(A_1 + B_1))$. ◀

References

1 Nikhil Bansal and Makrand Sinha. k-forrelation optimally separates quantum and classical query complexity. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 1303–1316, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3406325.3451040.

2 Paul Beame and Sajin Koroth. On disperser/lifting properties of the index and inner-product functions. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.ITCS.2023.14.

3 Eric Blais, Li-Yang Tan, and Andrew Wan. An inequality for the Fourier spectrum of parity decision trees. *CoRR*, abs/1506.01055, 2015. arXiv:1506.01055.

4 Arkadev Chattopadhyay, Yuval Filmus, Sajin Koroth, Or Meir, and Toniann Pitassi. Query-to-communication lifting using low-discrepancy gadgets. *SIAM J. Comput.*, 50(1):171–210, 2021. doi:10.1137/19M1310153.

5 Arkadev Chattopadhyay, Michal Koucký, Bruno Loff, and Sagnik Mukhopadhyay. Simulation theorems via pseudo-random properties. *Comput. Complex.*, 28(4):617–659, 2019. doi:10.1007/S00037-019-00190-7.

6 Arkadev Chattopadhyay, Nikhil S. Mande, Swagato Sanyal, and Suhail Sherif. Lifting to parity decision trees via stifling. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 33:1–33:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.ITCS.2023.33.

7 Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, July 2006.

8 Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. *Theory Comput.*, 16:1–30, 2020. doi:10.4086/TOC.2020.V016A013.

9 Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. *SIAM J. Comput.*, 47(5):1778–1806, 2018. doi:10.1137/16M1082007.

10 Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. *SIAM J. Comput.*, 47(6):2435–2450, 2018. doi:10.1137/16M1059369.

11 Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. *SIAM J. Comput.*, 49(4), 2020. doi:10.1137/17M115339X.

12 Hamed Hatami, Kaave Hosseini, and Shachar Lovett. Structure of protocols for XOR functions. *SIAM J. Comput.*, 47(1):208–217, 2018. doi:10.1137/17M1136869.

13 Bruno Loff and Sagnik Mukhopadhyay. Lifting theorems for equality. In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, volume 126 of *LIPIcs*, pages 50:1–50:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICS.STACS.2019.50.

14 Shachar Lovett, Raghu Meka, Ian Mertz, Toniann Pitassi, and Jiapeng Zhang. Lifting with sunflowers. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 104:1–104:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ITCS.2022.104.

15 Frédéric Magniez, Ashwin Nayak, Miklos Santha, and David Xiao. Improved bounds for the randomized decision tree complexity of recursive majority. In Luca Aceto, Monika Henzinger, and Jiří Sgall, editors, *Automata, Languages and Programming*, pages 317–329, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. doi:10.1007/978-3-642-22006-7_27.

16 Toniann Pitassi and Robert Robere. Strongly exponential lower bounds for monotone computation. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 1246–1255. ACM, 2017. doi:10.1145/3055399.3055478.

17 Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Comb.*, 19(3):403–435, 1999. doi:10.1007/S004930050062.

18 Robert Robere, Toniann Pitassi, Benjamin Rossman, and Stephen A. Cook. Exponential lower bounds for monotone span programs. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 406–415. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.51.

19 Miklos Santha. On the Monte Carlo Boolean decision tree complexity of read-once formulae. In *Proceedings of the Sixth Annual Structure in Complexity Theory Conference, Chicago, Illinois, USA, June 30 - July 3, 1991*, pages 180–187. IEEE Computer Society, 1991. doi:10.1109/SCT.1991.160259.

20 Xiaodi Wu, Penghui Yao, and Henry S. Yuen. Raz-McKenzie simulation with the inner product gadget. *Electron. Colloquium Comput. Complex.*, TR17-010, 2017. URL: https://eccc.weizmann.ac.il/report/2017/010.

Catalytic Communication

Edward Pyne
MIT, Cambridge, MA, USA

Nathan S. Sheffield
MIT, Cambridge, MA, USA

William Wang
MIT, Cambridge, MA, USA

Abstract

The study of space-bounded computation has drawn extensively from ideas and results in the field of communication complexity. Catalytic Computation (Buhrman, Cleve, Koucký, Loff and Speelman, STOC 2013) studies the power of bounded space augmented with a pre-filled hard drive that can be used non-destructively during the computation. Presently, many structural questions in this model remain open. Towards a better understanding of catalytic space, we define a model of *catalytic communication complexity* and prove new upper and lower bounds.

In our model, Alice and Bob share a blackboard with a tiny number of free bits, and a larger section with an arbitrary initial configuration. They must jointly compute a function of their inputs, communicating only via the blackboard, and must always reset the blackboard to its initial configuration. We prove several upper and lower bounds:

1. We characterize the simplest nontrivial model, that of one bit of free space and three rounds, in terms of $\mathbb{F}_2$ rank. In particular, we give natural problems that are solvable with a minimal-sized blackboard that require near-maximal (randomized) communication complexity, and vice versa.
2. We show that allowing constantly many free bits, as opposed to one, allows an exponential improvement on the size of the blackboard for natural problems. To do so, we connect the problem to existence questions in extremal graph theory.
3. We give tight connections between our model and standard notions of non-uniform catalytic computation. Using this connection, we show that with an arbitrary constant number of rounds and bits of free space, one can compute all functions in TC^0.

We view this model as a step toward understanding the value of filled space in computation.

2012 ACM Subject Classification Theory of computation → Communication complexity

Keywords and phrases Catalytic computation, Branching programs, Communication complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.79

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2024/188/

Acknowledgements We thank Ryan Williams for guidance and suggesting the question of catalytic communication complexity, and Carl Schildkraut for helpful discussion about Ruzsa–Szemerédi graphs.

1 Introduction

Communication complexity has proven an essential tool in analyzing the power of *space* in computation. For the well-studied problem of derandomizing space-bounded computation, i.e. proving $\mathsf{BPL} = \mathsf{L}$, the frontier pseudorandom generators of [36, 31] are analyzed by considering the space-bounded algorithm as a communication protocol. There has been extensive work analyzing restricted classes of space-bounded algorithms, again relying on this connection [9, 40, 16]. Other works have tightly characterized the space required to solve fundamental problems such as estimating the bias of a coin [10, 7, 8], using sophisticated measures of information complexity.


© Edward Pyne, Nathan S. Sheffield, and William Wang;
licensed under Creative Commons License CC-BY 4.0



Concurrently, a new model of bounded-space computation known as *catalytic* computation was introduced by Buhrman, Cleve, Koucký, Loff and Speelman [13], and used to solve fundamental computational problems more efficiently. In the Catalytic Logspace (CL) model, an algorithm receives an n-bit input, $O(\log n)$ bits of standard working space, and an auxiliary $\mathsf{poly}(n)$ bit catalytic tape τ. This tape has an arbitrary initial configuration, and must be reset to that starting configuration at the end of the computation. Despite a possible intuition that such a tape would not be useful, they showed that CL is likely to be *strictly* stronger than L – in particular, it contains logspace-uniform TC^1, and thus nondeterministic logspace (NL). Recently, Cook and Mertz used catalytic algorithms to show that the tree-evaluation problem, a candidate problem for separating L and P, can in fact be solved by an algorithm running in space $O(\log n \log\log n)$, contradicting long-standing prior beliefs [18, 19, 21]. Due to the striking power of this model, there have been many followup papers studying its structure [22, 15, 28, 39]. In the other direction, [17] used tools from the space-bounded literature (in particular, communication bottlenecks) to unconditionally *derandomize* CL. However, many basic questions remain open – it is consistent with current knowledge that $\mathsf{P} \subseteq \mathsf{CL}$, or that $\mathsf{CL} \subseteq \mathsf{NC}^2$, and there is no conditional evidence in either direction.

We aim to understand the power of access to a full memory by developing and studying the phenomenon in the setting of *communication complexity* – to what extent can such un-erasable extra memory be useful in exchanging information between two (computationally unbounded) parties?

1.1 Our Contribution: Catalytic Communication Complexity

We develop a natural model of catalytic communication complexity and prove several results, including maximal separations from the standard model of (randomized) communication complexity, a characterization of efficient protocols for the equality function in terms of extremal graph theory, and a strong equivalence between certain settings of the model and (nonuniform) catalytic computation itself.

Our model is defined as follows. There are two parties, Alice and Bob (perhaps graduate students coming to work on alternating days), sharing a single blackboard. Each party is computationally unbounded, but has no persistent memory between days, except what is written on the blackboard. Unfortunately, all but a tiny corner of the blackboard is currently full of someone else's important research, with instructions not to erase. Are Alice and Bob limited to sharing information through the tiny blank space, or can they do better by temporarily modifying the "Do Not Erase" section in such a way that still allows them to restore it when they're done? If they want to compute some function together, how big does the blackboard need to be, and how many days will they need? Such a model naturally captures the flow of information in a catalytic algorithm. The formal definition is as follows:

▶ **Definition 1** (Catalytic Communication Protocol). *Fix a function $f : \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$.*[1] *A **catalytic protocol computing f with r rounds, s bits of clean space, and c bits of catalytic space** consists of r many transition functions and one output function,*

$$A_1 : \{0,1\}^{n_a} \times \{0,1\}^c \times \{0,1\}^s \to \{0,1\}^c \times \{0,1\}^s$$

$$B_2 : \{0,1\}^{n_b} \times \{0,1\}^c \times \{0,1\}^s \to \{0,1\}^c \times \{0,1\}^s$$

[1] We will typically be concerned with the case $n_a = n_b = n$, but will also consider functions with asymmetric input lengths

$\dots$

$A_r : \{0,1\}^{n_a} \times \{0,1\}^c \times \{0,1\}^s \to \{0,1\}^c \times \{0,1\}^s$

$B_{out} : \{0,1\}^{n_b} \times \{0,1\}^c \times \{0,1\}^s \to \{0,1\}.$

(If r is even, we will instead have $B_r : \{0,1\}^{n_b} \times \{0,1\}^c \times \{0,1\}^s \to \{0,1\}^c \times \{0,1\}^s$ and $A_{out} : \{0,1\}^{n_a} \times \{0,1\}^c \times \{0,1\}^s \to \{0,1\}$.) For the protocol to be valid, for any $\tau \in \{0,1\}^c$, $x \in \{0,1\}^{n_a}$, $y \in \{0,1\}^{n_b}$, letting $A_i^{(x)} = A_i(x, \cdot)$ and $B_i^{(y)} = B_i(y, \cdot)$, these functions must satisfy

$$A_r^{(x)} \circ \dots \circ B_2^{(y)} \circ A_1^{(x)}(\tau, 0^s) = (\tau, w)$$

for some w with

$$B_{out}^{(y)}(\tau, w) = f(x, y).$$

In words, Alice and Bob's blackboard consists of c bits of arbitrarily-initialized catalytic space and s bits of initially-empty clean space. In a given round, the active party modifies the blackboard according to some arbitrary function of its current contents, the input visible to the active party, and the current timestep, then passes the blackboard to the other party. At the end of the protocol, the catalytic part of the blackboard must have been reset to its starting configuration, and the party with the blackboard must be able to announce the answer. Readers familiar with catalytic computing may observe that such a protocol could be described as an **amortized bipartite branching program**; this interpretation is the motivation for the particulars of our definition, and will be discussed in more detail in Section 6.

This model induces a corresponding measure of communication complexity:

▶ **Definition 2** (Catalytic Communication Complexity). *The **r-round s-clean catalytic communication complexity** of a function $f : \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$, denoted $\mathsf{CC}_{r,s}(f)$, is the minimum value of c such that f has an r-round catalytic protocol with s bits of clean space and c bits of catalytic space.*

The following three propositions represent simple observations about catalytic communication, demonstrating that this complexity measure is both well-defined and interesting. We give only the statements here; the associated proofs can be found in Section A. First note that with fewer than 3 rounds, the catalytic space is provably useless – in particular, this means that $\mathsf{CC}_{1,s}(f)$ and $\mathsf{CC}_{2,s}(f)$ are not well-defined for every f, since there may be no amount of catalytic space that suffices:

▶ **Proposition 3.** *Any function computable by a catalytic protocol with 1 or 2 rounds is computable by such a protocol with no catalytic space (i.e. there exists a memoryless protocol with the same amount of clean space, so without loss of generality only the clean space is used).*

For $r \geq 3$, $s \geq 1$, however, with enough catalytic space it is possible to compute any function, so $\mathsf{CC}_{r,s}(f)$ is always well-defined:

▶ **Proposition 4.** *Every function $f : \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$ has a 3-round catalytic protocol with 1 bit of clean space and $2^{\min(n_a, n_b)}$ bits of catalytic space.*

Proposition 4 represents an upper bound on catalytic communication complexity in general, but it is far from tight for some functions. Recall the inner product function (over $\mathbb{F}_2$) $\mathsf{IP}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, defined as $\mathsf{IP}_n(x,y) = \langle x, y\rangle$. We show that IP can be computed with one bit of clean space and n bits of catalytic space:

▶ **Proposition 5.** *We have $\mathsf{CC}_{3,1}(\mathsf{IP}_n) \leq n$, i.e. the inner product function has a 3-round catalytic protocol with 1 bit of clean space and n bits of catalytic space.*

Note that this protocol is roughly as efficient as a catalytic protocol can possibly be: as long as f is left-injective (i.e. no two values of x result in the same function $f(x,\cdot)$), it is clear for information theoretic reasons that the catalytic space cannot be made much less than n bits, and certainly the clean space required for any non-constant f must be at least 1 bit – so inner product is essentially the *easiest possible* function for this model. This is in contrast to standard notions of communication complexity: inner product has, up to constant factors, maximal deterministic and randomized communication complexity [41]. Such a separation even for a single bit of clean space indicates that the "picture" of catalytic communication can diverge widely from that of standard models.

Furthermore, unlike deterministic and randomized communication complexity, there does not appear to be a direct counting argument showing that a random function requires any amount of catalytic space larger than the trivial bound of n. Despite this barrier, we give a tight characterization of protocols with one bit of clean space and three rounds. As our first main result, we show that $\mathsf{CC}_{3,1}$ is characterized up to a constant factor by $\mathbb{F}_2$ rank:

▶ **Theorem 6.** *For any $f\colon \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$,*

$$\frac{\mathrm{rk}(f)}{6} - o(1) \leq \mathsf{CC}_{3,1}(f) \leq \mathrm{rk}(f),$$

where $\mathrm{rk}(f)$ denotes the rank of the communication matrix over $\mathbb{F}_2$.

The proof of the upper bound of Theorem 6 is an explicit protocol, while the lower bound involves an application of Harper's vertex-isoperimetric theorem on an appropriate subspace of the Boolean hypercube. We remark that several standard communication complexity measures are suspected to be closely controlled by appropriate notions of the rank of the communication matrix, but this case is unusual in that it is the rank over $\mathbb{F}_2$, as opposed to $\mathbb{R}$, that is relevant.

This establishes catalytic communication complexity as a very different measure from standard ones. In particular, if one compares catalytic communication complexity against standard randomized communication complexity[2], there are maximal separations for natural functions in *both* directions. Inner product has a maximally efficient 3-round 1-clean catalytic communication protocol but no standard randomized communication protocol with communication less than the trivial $\Omega(n)$. On the other hand, the equality function EQ_n has $\mathsf{CC}_{3,1}(\mathsf{EQ}_n) = \Omega(2^n)$ but randomized communication complexity $O(1)$ [41].

The lower bound approach we use for $\mathsf{CC}_{3,1}$ does not extend to $\mathsf{CC}_{3,2}$, i.e. protocols with a single extra bit of clean space. This is not simply a consequence of technical limitations – we show that for equality, allowing a single additional bit of clean space enables an exponentially more efficient protocol:

[2] Comparing against *deterministic* standard communication complexity, we would instead find that catalytic communication is strictly stronger, in the sense that $\log(\mathsf{CC}_{3,1}(f))$ is always at most the deterministic communication complexity of f.

▶ **Theorem 7.** *We have that* $\mathsf{CC}_{3,2}(\mathsf{EQ}_n) \leq O(n \log n)$.

Our protocol for equality follows from a connection to a new variant we propose of the well-known Ruzsa–Szemerédi problem in extremal graph theory. We demonstrate that efficient 3-round catalytic protocols for equality can be obtained generically from constructions of "full Ruzsa–Szemerédi graphs", a special case of Ruzsa–Szemerédi graphs. We show that the current frontier Ruzsa–Szemerédi construction can be modified to satisfy our stronger condition, and hence obtain an efficient protocol. We also show a reverse implication – that is, that efficient protocols imply dense Ruzsa–Szemerédi constructions – enabling us to obtain nontrivial lower bounds for the complexity of equality:

▶ **Theorem 8.** *For any constant* s*, we have* $\mathsf{CC}_{3,s}(\mathsf{EQ}_n) \geq n + \Omega(\log^*(n))$.

In addition to this unconditional lower bound, the connection to Ruzsa–Szemerédi graphs also gives a barrier result: improving our upper bound to $\mathsf{CC}_{3,O(1)}(\mathsf{EQ}_n) = O(n)$ would yield (among other things) a polynomial query *lower* bound for testing monotonicity of Boolean functions over general posets.

Towards obtaining lower bounds stronger than $n + \Omega(\log^* n)$, we propose considering protocols for the **indexing problem** $\mathsf{IND}_n : \left([2^n] \times \{0,1\}^{2^n}\right) \to \{0,1\}$, $\mathsf{IND}_n(x, y) = y[x]$. Here, Alice's input is a length-n bitstring representing an index, and Bob's input is a length-2^n bitstring, with the output of the function being the bit of Bob's input corresponding to Alice's index. Another description of this problem is that Alice is given some n-bit input, and Bob is given the truth table of some arbitrary boolean function on n-bit inputs, and they must compute the evaluation of Bob's function on Alice's input. It is clear from this phrasing that proving a complexity lower bound on any function in terms of Alice's input length would require proving at least as strong a lower bound on IND_n, as any other function is a special case. We find a graph theoretic characterization of protocols for this problem, which allows us to show a non-trivial lower bound via the Kővári-Sós-Turán theorem.

▶ **Theorem 9.** *We have* $\mathsf{CC}_{3,s}(\mathsf{IND}_n) \geq (1 + \varepsilon)n$*, for some constant* ε *depending on* s.

While we suspect Theorem 9 is far from tight, it remains open to show super-linear lower bounds even for 2 clean bits.

Finally, we study protocols with more rounds. We note that our measure of catalytic communication complexity corresponds directly to the minimum amount of amortization required in an amortized bipartite branching program of bounded length. Since an ordinary branching program is in particular a bipartite branching program, this immediately lets us translate known results from nonuniform catalytic computing to show statements such as $\mathsf{CC}_{\mathsf{poly}(n),O(1)}(f) \leq O(n)$ for all f. By a more specialized analysis of Buhrman, Cleve, Koucký, Loff and Speelman's algebraic proof of $\mathsf{TC}^1 \subseteq \mathsf{CL}$, we are also able to show the following:

▶ **Theorem 10.** *Let* $f\colon \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$ *be any function computable by a size* S*, depth* d *majority circuit with arbitrary input preprocessing – that is, there exists a depth-*d *circuit* C *composed of* S *many majority gates, and arbitrary functions* g, h*, such that* $f(x, y) = C(g(x), h(y))$*. Then,* $\mathsf{CC}_{4^d,1}(f) \leq \mathsf{poly}(S, 2^d)$.

This implies in particular that any TC^0 function family has constant-round, 1-clean catalytic protocols with $\mathsf{poly}(n)$ catalytic space.

We view these results as indicating that catalytic communication complexity is both an interesting model in its own right, and one that can shed light on reusing space more generally. We remark that our model addresses a question raised in prior work on variants of communication complexity, which we now discuss.

1.2 Related Work

Identifying an interesting communication analogue of catalytic computing was posed as an open problem by Arunachalam and Podder in a paper on space-bounded communication [2]. Space-bounded communication complexity was first studied by Brody, Chen, Papakonstantinou, Song and Sun, who proposed a model in which the two parties in a standard communication protocol are in addition required to compress each of their states into a small number of bits after each message is sent [11]. Subsequent work by Papakonstantinou, Scheder, and Song considered a one-way model, in which Alice (a party with unbounded memory) sends messages of length s to Bob (a party with zero, or only constant, memory) – they showed that when Bob is memoryless, this model can be characterized in terms of a combinatorial notion of rectangle overlays, and when Bob has only 5 available memory states and $s = \mathsf{polylog}(n)$ this model computes exactly PSPACE^{cc} [37]. Arunachalam and Podder suggested instead letting both Alice and Bob be memoryless, and measuring complexity in terms of the size of a block of memory they pass back and forth [2]. They noted that this model aligns closely with bipartite branching program complexity, and that variants could capture the aforementioned models of Brody, Chen, Papakonstantinou, Song and Sun [11] and Papakonstantinou, Scheder and Song [37], as well as the "garden hose" model of Buhrman, Fehr, Schaffner and Speelman [14]. Our model can be thought of as a version of Arunachalam and Podder's memoryless framework where the size of the block of memory passed back and forth is *amortized*, in a sense analogous to the way workspace usage is amortized across inputs in a catalytic algorithm.

2 Preliminaries

We will by convention denote a setting of the catalytic portion of the blackboard by τ (for "transparent registers"), a setting of the initially-clean portion by ω (for "work registers"), and a setting of the entire blackboard by γ (for no particular reason). We will let $x \in \{0,1\}^{n_a}$ be the input given to Alice, and $y \in \{0,1\}^{n_b}$ be the input given to Bob. We now define a few terms and notations which appear in the statements and proofs.

▶ **Definition 11.** *For any n, we denote by IP_n the inner product function over $\mathbb{F}_2^n$. That is,*

$$\mathsf{IP}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\},$$

$$\mathsf{IP}(x,y) = \sum_{i\in[n]} x_i y_i \mod 2.$$

▶ **Definition 12.** *For any n, we denote by EQ_n the equality function on n-bit inputs. That is,*

$$\mathsf{EQ}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\},$$

$$\mathsf{EQ}(x,y) = \begin{cases} 1 \text{ if } x = y \\ 0 \text{ if } x \neq y \end{cases}.$$

▶ **Definition 13.** *For any n, we denote by IND_n the function taking an n-bit index and a 2^n-bit bitstring to the value of that bitstring at that index. That is,*

$$\mathsf{IND}_n : [2^n] \times \{0,1\}^{2^n} \to \{0,1\},$$

$$\mathsf{IND}_n(x,y) = y[x].$$

▶ **Definition 14.** *For a function $f : \{0,1\}^{n_a} \to \{0,1\}^{n_b} \to \{0,1\}$, let the **communication matrix** of f be the $2^n \times 2^n$ matrix M where $M_{x,y} = f(x,y)$, and let $\mathrm{rk}(f)$ be the rank of M over $\mathbb{F}_2$.*

For a function on two inputs $f\colon X, Y \to Z$, we will use $f(x,\cdot)$ to denote the application $f(x,\cdot)\colon Y \to Z$. We say a function $f\colon X, Y \to Z$ is **left-injective** if there do not exist $x \neq x' \in X$ such that $f(x,\cdot)$ and $f(x',\cdot)$ are identical functions. For integers n, we will write $[n]$ to denote $\{1,\dots,n\}$. To denote to a multiset with elements $a_1,\dots,a_n$, we will use the "bag" notation $\wr a_1,\dots,a_n\int$.

3 Characterizing 3-Round 1-Clean Complexity

In this section, we prove Theorem 6.

▶ **Theorem 6.** *For any $f\colon \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$,*

$$\frac{\mathrm{rk}(f)}{6} - o(1) \leq \mathsf{CC}_{3,1}(f) \leq \mathrm{rk}(f),$$

where $\mathrm{rk}(f)$ denotes the rank of the communication matrix over $\mathbb{F}_2$.

We first give the upper bound, showing that any function with small rank has an efficient catalytic protocol.

▶ **Lemma 15.** *For any $f\colon \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$, we have $\mathsf{CC}_{3,1}(f) \leq \mathrm{rk}(f)$.*

Proof. By definition of rank, the rows of f's communication matrix span a $\mathrm{rk}(f)$-dimensional subspace of $\mathbb{F}_2^n$. Both parties fix ahead of time a basis $v_1,\dots v_{\mathrm{rk}\, f}$ for this subspace, and define a protocol as follows:

1. On the first round, Alice finds $T \subseteq [\mathrm{rk}\, f]$ such that $\bigoplus_{i\in T} v_i$ is the xth row of the communication matrix (i.e., is equal to the truth table of the function $y \mapsto f(x,y)$), and sets the catalytic portion of the blackboard to $\tau \oplus 1_{i\in T}$.
2. Bob computes $\bigoplus_{i\in[\mathrm{rk}\, f]} \tau_i v_i$, and writes the yth entry of the result to the clean portion.
3. Alice XORs out the same string she XORed in, resetting the catalytic portion.
4. Bob finds the new yth entry of $\bigoplus_{i\in[\mathrm{rk}\, f]} \tau_i v_i$, and outputs its XOR with the clean bit.

The correctness of this protocol is straightforward. After the second round of the protocol, Bob sets the clean bit to

$$b_1 = \left(\bigoplus_{i\in[\mathrm{rk}\, f]} (\tau_i \oplus 1_{i\in T}) v_i \right)_y,$$

and so his final output will be

$$b_1 \oplus \left(\bigoplus_{i\in[\mathrm{rk}\, f]} \tau_i v_i \right)_y = \left(\bigoplus_{i\in T} v_i \right)_y = f(x,y),$$

as desired. The resource consumption is immediate from the protocol definition. ◀

We would like to show that it is impossible to do substantially better than the above protocol. Note that the protocol described in Lemma 15 follows a simple pattern: Alice modifies only the catalytic portion of the blackboard, performing some bijective map on all possible settings, Bob uses only the clean portion of the blackboard to "remember" one bit

about the result, and then Alice undoes her bijection and Bob makes his output decision based on his remembered bit in addition to that initial catalytic configuration. For the purposes of showing lower bounds, it would be helpful first to claim that without loss of generality all 3-round protocol must be of that form. However, this is not *quite* true. For one, Alice could use the clean portion of the blackboard to send Bob an additional bit of information about her input on the first round, which is provably helpful sometimes. Another rather more substantial concern is that there's room for "amortization" in the amount of information that Bob remembers: it could, for instance, be possible that, on some particular settings of Alice's input x and the initial catalytic setting C, Alice can on her first round encode all of the catalytic information in a small prefix of the tape – in such cases, Bob could use the rest to send Alice a large amount of information about *his* input, which would then inform her in choosing a single bit send him back in the clean space of the final setting. However, we expect that this latter sort of behaviour should only happen rarely, since most initial catalytic settings can't be compressed. In the full version, we address these concerns formally, guaranteeing that any 3-round 1-clean catalytic protocol can be converted into a protocol that *almost* follows the simple pattern we noted. Specifically, we show the following:

▶ **Lemma 16.** *Every left-injective function $f : \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$ with $\mathsf{CC}_{3,1}(f) = c$ has a protocol of the following form, which we'll call a* ***mostly-one-way-catalytic protocol****:*

1. *For every $x \in \{0,1\}^{n_a}$, Alice has an injective function $\alpha^{(x)} : \{0,1\}^c \to \{0,1\}^c \times \{0,1\}^3$.*
2. *For every $y \in \{0,1\}^{n_b}$, Bob has two functions, $\beta^{(y)}_{rem} : \{0,1\}^c \times \{0,1\}^3 \to \{0,1\}$ and $\beta^{(y)}_{out} : \{0,1\}^c \times \{0,1\} \to \{0,1\}$.*
3. *Call a pair $(x,\tau) \in \{0,1\}^{n_a} \times \{0,1\}^c$* ***bad*** *if, for some $y \in \{0,1\}^{n_b}$, we have*

$$\beta^{(y)}_{out}(\tau, \beta^{(y)}_{rem}(\alpha^{(x)}(\tau))) \neq f(x,y).$$

Then, at most 2^{c+1} many pairs are bad.

Additionally, we may assume that, for every τ, y, we have $\beta^{(y)}_{out}(\tau,0) \neq \beta^{(y)}_{out}(\tau,1)$.

This observation will allow us to associate a communication protocol with a structured collection of vectors in $\mathbb{F}_2^{2^{n_b}}$, whose size we can bound using standard combinatorial facts.

Proof of Theorem 6. The upper bound was shown in Lemma 15; it remains to show the lower bound. Note that removing duplicate rows from the communication matrix of a function can change neither its rank, nor the catalytic communication complexity (since a protocol for the function on the larger domain could simply treat several of its inputs identically). So, it suffices to show the claim for left-injective functions. Also, note that, given the $o(1)$ term in the statement, it suffices to only consider the case where $\mathrm{rk}(f) \geq 1000$. We fix an arbitrary left-injective $f\colon \{0,1\}^{n_a} \times \{0,1\}^{n_b}$ with $\mathrm{rk}(f) \geq 1000$, and assume for contradiction that $\mathsf{CC}_{3,1}(f) < \mathrm{rk}(f)/6$.

By Lemma 16, we have a mostly-one-way-catalytic protocol for f with $c < \mathrm{rk}(f)/6$. We will use this protocol to define three multi-subsets $U, V, W \subseteq \{0,1\}^{n_b}$. For every $\gamma \in \{0,1\}^{c+3}$, let $u_\gamma \in \{0,1\}^{n_b}$ be the truth table of $y \mapsto \beta^{(y)}_{\mathrm{rem}}(\gamma)$, and let $U = \{\!\{u_\gamma \colon \gamma \in \{0,1\}^{c+3}\}\!\}$. For every $\tau \in \{0,1\}^c$, let $v_\tau \in \{0,1\}^{n_b}$ be the truth table of $y \mapsto \beta^{(y)}_{\mathrm{out}}(\tau,0)$, and let $V = \{\!\{v_\tau \colon \tau \in \{0,1\}^c\}\!\}$. Finally, for every bad $(x,\tau) \in \{0,1\}^{n_a} \times \{0,1\}^c$, let $w_{(x,\tau)}$ be the truth table of $y \mapsto f(x,y) \oplus \beta^{(y)}_{\mathrm{out}}(\tau,0)$, and let $W = \{\!\{w_{(x,\tau)} \colon (x,\tau) \text{ is bad}\}\!\}$.

Note that $|U| = 2^{c+3}$, $|V| = 2^c$, and $|W| \leq 2^{c+1}$. The idea of the proof will be to show that $U \cup W$ contains the neighbours in an appropriate Boolean hypercube of every element of V (with appropriate multiplicity) – since small subsets of the hypercube have large vertex

boundary compared to their volume, but we know that $|U| + |W|$ is only a constant factor larger than $|V|$, this will allow us to give a lower bound on c. This approach is formalized as follows.

Imagine placing red and blue pebbles on $\mathbb{F}_2^{n_b}$, such that each element of $\mathbb{F}_2^{n_b}$ gets a number of red pebbles equal to the number of times it appears in $U \cup W$, and a number of blue pebbles equal to the number of times it appears in V. We claim that the following must hold:

▷ Claim 17. For any such pebbling constructed from a valid protocol, for any $k \in \mathbb{N}$, if $v \in \mathbb{F}_2^{n_b}$ has at least k blue pebbles and $r \in \mathbb{F}_2^{n_b}$ is a row of f's communication matrix (i.e. r is the truth table of $y \mapsto f(x, y)$ for some x), then $v \oplus r$ has at least k red pebbles.

Proof of Claim 17. Because v has at least k blue pebbles, there are at least k distinct initial catalytic configurations $\tau_1, \ldots, \tau_k \in \{0,1\}^c$ such that $v_{\tau_i} = v$. Let x be the input such that r is the truth table of $y \mapsto f(x, y)$. For every bad τ_i, we have $w_{(x,\tau_i)} = r \oplus v_{\tau_i} = v \oplus r$, so each of these will contribute a red pebble to $v \oplus r$. Then, since $\alpha^{(x)}$ is injective, each of $\alpha^{(x)}(\tau_1), \ldots, \alpha^{(x)}(\tau_k)$ must be distinct. So, to obtain the remainder of the red pebbles, we'll show that whenever (x, τ_i) is not bad, $u_{\alpha^{(x)}(\tau_i)} = v \oplus r$.

Note that r is the truth table of $y \mapsto f(x, y) \oplus v$, that $v = v_{\tau_i}$ is the truth table of $y \mapsto \beta^{(y)}_{\text{out}}(\tau_i, 0)$, and that $u_{\alpha^{(x)}(\tau_i)}$ is the truth table of $y \mapsto \beta^{(y)}_{\text{rem}}(\alpha^{(x)}(\tau_i))$. So, to conclude that $r = v \oplus u_{\alpha^{(x)}(\tau_i)}$, we need to show that, for all y, we have $f(x, y) = \beta^{(y)}_{\text{out}}(\tau_i, 0) \oplus \beta^{(y)}_{\text{rem}}(\alpha^{(x)}(\tau_i))$.

Since (x, τ_i) is not bad, we have $f(x, y) = \beta^{(y)}_{\text{out}}(\tau, \beta^{(y)}_{\text{rem}}(\alpha^{(x)}(\tau)))$. If $\beta^{(y)}_{\text{rem}}(\alpha^{(x)}(\tau)) = 0$, this gives $f(x, y) = \beta^{(y)}_{\text{out}}(\tau, 0) = \beta^{(y)}_{\text{out}}(\tau, 0) \oplus 0$, so the claim holds. If, on the other hand, $\beta^{(y)}_{\text{rem}}(\alpha^{(x)}(\tau)) = 1$, we have $f(x, y) = \beta^{(y)}_{\text{out}}(\tau, 1)$. But then, since we always have $\beta^{(y)}_{\text{out}}(\tau, 0) \neq \beta^{(y)}_{\text{out}}(\tau, 1)$, we know that $\beta^{(y)}_{\text{out}}(\tau, 1) = \beta^{(y)}_{\text{out}}(\tau, 0) \oplus 1$. ◁

This will be sufficient to prove that there must be many blue pebbles, contradicting the assumption that c is small. Consider the graph on $\mathbb{F}_2^{n_b}$ where an edge (u, v) exists whenever $u \oplus v$ is a row of f's communication matrix. This graph consists of $2^{n_b}/2^{\text{rk}(f)}$ many identical connected components. If we fix some subset $X \subseteq \{0,1\}^{n_a}$ such that the corresponding rows of the communication matrix form a basis for the row-space, and consider only the edges generated by those rows, each of these connected components will be isomorphic to the $\text{rk}(f)$-dimensional hypercube. At least one of these connected components must contain a blue pebble; we will restrict our attention to one such connected component.

Claim 17 ensures that each vertex of this hypercube has at least as many red pebbles as the maximum number of blue pebbles among its neighbours. Consider the set S of vertices with at least one blue pebble. All vertices adjacent to a vertex of S must have at least one red pebble; we claim that there are many such vertices.

▷ Claim 18. For any subset $S \subseteq \mathbb{F}_2^{\text{rk}(f)}$ with $|S| \leq 2^{\text{rk}(f)/6}$, we have $|N(S)| > 10 \cdot |S|$, where $N(S)$ denotes the set of all Hamming neighbours of elements of S.

Proof of Claim 18. Take S to be a set of the smallest possible size such that $|N(S)| \leq 10 \cdot |S|$. Harper's theorem states that, among all subsets of the Boolean hypercube of a given size, the vertex boundary is minimized by a Hamming ball. That is, for any ℓ, if $|S| = \sum_{i=0}^{\ell} \binom{\text{rk}(f)}{i}$, then $|N(S) \setminus S| \geq \binom{\text{rk}(f)}{\ell+1}$ [29, 6]. Let ℓ be the radius of the smallest Hamming sphere larger than our fixed set S; i.e., the smallest integer such that $|S| \leq \sum_{i=0}^{\ell} \binom{\text{rk}(f)}{i}$. Since $\min_{S:\,|S|=s} \left(\frac{|N(S)|}{|S|} \right)$ is monotonically nonincreasing in s for small s[3], the surface-area-to-volume ratio of this Hamming sphere lower bounds that of S:

[3] As long as s is small enough that a Hamming ball of that size can't have vertex boundary less than

$$\frac{|N(S)|}{|S|} \geq \frac{\binom{\mathrm{rk}(f)}{\ell+1}}{\sum_{i=0}^{\ell}\binom{\mathrm{rk}(f)}{i}}.$$

By minimality of S, we know that $\sum_{i=0}^{\ell'}\binom{\mathrm{rk}(f)}{i} \leq \binom{\mathrm{rk}(f)}{\ell'+1}$ for all $\ell' < \ell$, since otherwise the Hamming ball of radius ℓ' would be a smaller example. So, $\sum_{i=0}^{\ell}\binom{\mathrm{rk}(f)}{i} \leq \binom{\mathrm{rk}(f)}{\ell} + \sum_{i=0}^{\ell-1}\binom{\mathrm{rk}(f)}{i} \leq 2\binom{\mathrm{rk}(f)}{\ell}$. Thus,

$$\frac{|N(S)|}{|S|} \geq \frac{\binom{\mathrm{rk}(f)}{\ell+1}}{2\binom{\mathrm{rk}(f)}{\ell}} = \frac{\frac{\mathrm{rk}(f)!}{(\ell+1)!(\mathrm{rk}(f)-\ell-1)!}}{\frac{2\,\mathrm{rk}(f)!}{\ell!(\mathrm{rk}(f)-\ell)!}} = \frac{(\mathrm{rk}(f)-\ell)}{2(\ell+1)}.$$

Since we assumed that $\frac{|N(S)|}{|S|} \leq 10$, this gives $\ell \geq \frac{\mathrm{rk}(f)-19}{20}$. Now, we have

$$|S| \geq \sum_{0}^{\ell-1}\binom{\mathrm{rk}(f)}{i} \geq \binom{\mathrm{rk}(f)}{\ell-1} \geq 2^{\mathrm{rk}(f)\cdot H\left(\ell/\,\mathrm{rk}(f)\right)-\log\left(\mathrm{rk}(f)\right)} \geq 2^{\mathrm{rk}(f)\cdot H\left(1/20-1/\,\mathrm{rk}(f)\right)-\log\left(\mathrm{rk}(f)\right)},$$

where $H\colon p \mapsto -p\log(p) - (1-p)\log(1-p)$ is the binary entropy function. For all values of $\mathrm{rk}(f) \geq 1000$, we have $\mathrm{rk}(f)H\big(1/20 - 1/\,\mathrm{rk}(f)\big) - \log\big(\mathrm{rk}(f)\big) > \frac{\mathrm{rk}(f)}{6}$, so this implies that $|S| > 2^{\left(\mathrm{rk}(f)/6\right)}$. ◁

We can't immediately a derive contradiction from Claim 17 and Claim 18, because V is a multiset (i.e. some vertices may have more than one red pebble). However, this is not a serious obstacle.

▷ Claim 19. Suppose a pebbling satisfies the conditions of Claim 17 – i.e., that $v \oplus r$ always has at least as many blue pebbles as v has red pebbles for any row r of the communication matrix. Then, if we remove one blue pebble from every vertex with at least one blue pebble, and one red pebble from any vertex with at least one red pebble, that condition still holds.

Proof. Consider any v and any r. Before removal, $v \oplus r$ had at least as many blue pebbles as v had red pebbles. If v had 0 blue pebbles, then after removal this will remain the case, so $v \oplus r$ will still be at least as pebbled. Otherwise, we will remove the same number of blue pebbles from v as we remove red pebbles from $v \oplus r$. ◁

If we remove a single blue pebble from every vertex with at least one blue pebble, and a single red pebble from every vertex with at least one red pebble, Claim 18 ensures that we will remove strictly more than ten times as many red pebbles as blue pebbles. After removing those pebbles, there are still at most 2^c vertices with blue pebbles, and we've just shown that Claim 17 still holds – so, we can repeatedly perform such removals until all pebbles are gone, always removing strictly more than ten times as many red pebbles as blue pebbles. This means that the graph overall has *strictly* more than ten times as many red pebbles as blue pebbles, contradicting the fact that $|U| + |W| = 2^{c+3} + 2^{c+1} = 10 \cdot 2^c = 10 \cdot |V|$. ◀

One interesting consequence of Theorem 6 is that the equality function, which has constant randomized communication complexity, requires the maximal $\Omega(2^n)$ blackboard size in this catalytic model. On the other hand, as noted, the inner product function can

twice the volume (which, as we show, is true here), one can observe that adding a new point at the boundary will always increase the surface-area-to-volume ratio.

be communicated with only n bits of catalytic space, despite requiring (up to constant factors) maximal standard communication complexity even with randomness [41]. Catalytic communication complexity seems to be quantifying a very different notion from standard communication complexity measures.

A natural question at this stage is whether our choice to restrict to only a single bit of clean space was roughly without loss of generality, or whether protocols with more clean space can look substantially different. Having more free space is definitely at least moderately helpful: for instance, equality can be computed with 2 bits of free space and $2 \cdot 2^{n/2}$ bits of catalytic space, by simply running two copies of the 1-bit protocol on the two halves of the inputs simultaneously, and having Bob output the AND of the answers. But one might suspect that, with only a little more clean space, there is not too much more that can be done – perhaps for any constant s it's possible to show that $\mathsf{CC}_{3,s}(\mathsf{EQ}_n) \geq 2^{\Omega(n)}$. As we will show in the next section, this intuition turns out to be false – even with two just bits of clean space, it's possible to communicate equality very efficiently.

4 The 3-Round Complexity of Equality

In this section, we prove the following upper bound on 3-round 2-clean catalytic complexity of equality.

▶ **Theorem 7.** *We have that* $\mathsf{CC}_{3,2}(\mathsf{EQ}_n) \leq O(n \log n)$.

The key to the proof of Theorem 7 comes from a long line of research on graphs representable as the union of many large induced matchings, known as Ruzsa–Szemerédi graphs. We will provide background on the Ruzsa–Szemerédi problem in Section 4.1, where we will also introduce a variant important for our application. In Section 4.2, we will demonstrate how any such construction can be generically converted into a catalytic protocol for equality, which will in particular prove Theorem 7.

In Section 4.3 we will demonstrate a reverse implication – that is, that efficient 3-round equality protocols give dense Ruzsa–Szemerédi protocols – which will allow us to state catalytic communication complexity lower bounds. We will show unconditionally that there exists no 3-round $O(1)$-clean protocol for equality using $n + O(1)$ bits of catalytic space, and show that finding any stronger upper bound than $O(n \log n)$ would require improved graph theoretic constructions that would in turn prove new *lower* bounds in areas such as property testing and streaming algorithms.

4.1 Background on the Ruzsa–Szemerédi Problem

We give the following definition:

▶ **Definition 20.** *A graph G is an (r, t)-**Ruzsa–Szemerédi graph** if there exists a partition of its edges into t sets of size r, such that each set constitutes an induced matching in G.*

In 1976, motivated by a problem of Brown, Erdős and Sós on 3-uniform hypergraphs containing no 6 vertices sharing 3 edges [12], Ruzsa and Szemerédi proposed studying such graphs, with the goal of finding examples where both r and t are large compared to number of vertices of G. They showed that Behrend's construction of sets without 3-term arithmetic progression [4] gives $\left(\frac{n}{2^{O(\sqrt{\log(n)})}}, \frac{n}{3}\right)$-Ruzsa–Szemerédi graphs, and used regularity methods to show that r and t cannot both be made $\Omega(n)$ [43]. The best known upper bounds still go through graph regularity: Fox's improved bounds for triangle removal imply that if $r = \Theta(n)$, then $t \leq \frac{n}{2^{\Omega(\log^*(n))}}$ [24].

Ruzsa–Szemerédi graphs have since seen several applications in computational and communication complexity [30, 5, 35]. In particular, constructions where $r = \Theta(n)$ and t is large have been used to obtain a number of lower bounds in streaming algorithms and property testing [23, 27, 33, 32, 3]. The best known lower bound on t when $r = \Theta(n)$ is due to Fischer, Lehman, Newman, Raskhodnikova, Rubinfeld and Samorodnitsky, giving a construction where $r = n/3$ and $t \geq n^{\Omega\left(\frac{1}{\log\log(n)}\right)}$ – eliminating the gap between this bound and the $t \leq \frac{n}{2^{\Omega(\log^*(n))}}$ upper bound is a major open problem in combinatorics [23].

For our application, we also give the following more restrictive definition:

▶ **Definition 21.** *A graph G is a (k,t)-**full Ruzsa–Szemerédi graph** if there exists a partition of its edges into t perfect matchings, and a partition of each of those matchings into k partial matchings, such that each partial matching is induced in G.*

Note that a (k,t)-full Ruzsa–Szemerédi graph is in particular a $(n/2k, kt)$-Ruzsa–Szemerédi graph, but that here we are imposing an additional constraint by requiring that the induced matchings can be grouped together to form perfect matchings in G. The construction presented by Fischer, Lehman, Newman, Raskhodnikova, Rubinfeld and Samorodnitsky is not a full Ruzsa–Szemerédi graph, but we show in the full version that it can easily be made so:

▶ **Lemma 22.** *As long as $n = 2\ell^\ell$ for some integer ℓ, there exists a $\left(3, n^{\Omega\left(\frac{1}{\log\log n}\right)}\right)$-full Ruzsa–Szemerédi graph.*

4.2 Equality Protocols from Full Ruzsa–Szemerédi Graphs

We now show that full Ruzsa–Szemerédi graphs can be generically converted to catalytic protocols.

▶ **Lemma 23.** *If there exists a (k,t)-full Ruzsa–Szemerédi graph on 2^c vertices, then there exists a 3-round, $\lceil\log(k)\rceil$-clean catalytic protocol for equality on $\lfloor\log(t)\rfloor$-bit inputs with c bits of catalytic space*[4]*.*

We first note that this will immediately give us our desired upper bounds on $\mathsf{CC}_{3,2}(\mathsf{EQ}_n)$.

Proof of Theorem 7 given Lemma 23. For some constant $0 < \delta < 1$, by Lemma 22, there exists a $(3, N^{\frac{\delta}{\log\log N}})$-full Ruzsa–Szemerédi graph on N vertices. Plugging this in to Lemma 23, we obtain a $\lceil\log(3)\rceil = 2$-clean catalytic protocol for equality on $\lfloor\log(t)\rfloor = \lfloor\delta\log(N)/\log\log(N)\rfloor$ bit inputs, requiring $\log(N)$ bits of catalytic space. Defining $n = \lfloor\log(t)\rfloor$, this is a 3-round 2-clean protocol with $O(n\log n)$ bits of catalytic space[5]. ◀

Proof of Lemma 23. Let G be such a graph, with vertices $V(G) = v_1, \ldots, v_{2^c}$, composed of perfect matchings $M_1 \sqcup \cdots \sqcup M_t = E(G)$. The full-Ruzsa–Szemerédi property gives, for each $i \in [t]$, a vertex partition $M_i^{(1)} \sqcup \cdots \sqcup M_i^{(k)} = V(G)$ such that the edges of M_i are the union

[4] Observe though, if one cares about such things, that unless that Ruzsa–Szemerédi graph can be constructed explicitly, the resulting protocol may not be computationally efficient. In order to get a computationally effective protocol from our construction, one would have to plug in an explicit code as opposed to just citing Gilbert-Varshanov.

[5] Note that this construction is only defined when both N is a power of two, and $N = 2\cdot\ell^\ell$ for some integer ℓ. However, if we let ℓ be the smallest power of two such that $2\cdot\ell^\ell \geq N$, and let $N' = 2\cdot\ell^\ell$, then note that $\log(N') \leq 4\cdot\log(N)$. So, for other values of n, Alice and Bob can simply pad their inputs with zeroes and run the equality protocol for a value of n where this is defined, losing only a constant factor in the amount of catalytic space required.

over j of the edges induced by $M_i^{(j)}$. Fix an arbitrary bijection $\pi\colon \{0,1\}^c \to \{v_1,\dots,v_N\}$, and arbitrary injections $\sigma\colon \{0,1\}^{\lfloor \log(t)\rfloor} \to \{M_1,\dots,M_t\}$, $\mu\colon [k] \to \{0,1\}^{\lceil \log k\rceil}$. The protocol is as follows.

1. $A_1(x,\tau,\omega) = (\pi^{-1}(v),\omega)$, where v is the neighbour of $\pi(\tau)$ in the matching $\sigma(x)$.
2. $B_2(y,\tau,\omega) = (\tau,\mu(w))$, where w is the unique value such that $\pi(\tau) \in \sigma(y)^{(w)}$.
3. $A_3(x,\tau,\omega) = (\pi^{-1}(v),\omega)$, where v is the neighbour of $\pi(\tau)$ in the matching $\sigma(x)$.
4. $B_{\text{out}}(y,\tau,\omega)$ accepts if and only if $\omega = \mu(w)$, where w is the unique value such that $\pi(\tau) \in \sigma(y)^{(w)}$.

Correctness of this protocol follows from the definition of a full Ruzsa–Szemerédi graph. If $x = y$, then the matchings $\sigma(x)$ and $\sigma(y)$ are the same. Since every edge of $\sigma(x)$ is induced by some part of the partition $\sigma(x)^{(1)},\dots,\sigma(x)^{(k)}$, this means that both endpoints of that edge belong to the same $\sigma(y)^{(w)}$, and so Bob will accept. If, on the other hand, $x \neq y$, then the two endpoints of an edge in $\sigma(x)$ cannot belong to the same $\sigma(y)^{(k)}$, because $\sigma(y)^{(k)}$ induces only the edges belonging to $\sigma(y)$. ◀

4.3 Lower Bounds on Equality Protocols

In this section, we will show that, as long as we are content with a not-necessarily-full Ruzsa–Szemerédi graph, the conversion in Lemma 23 can be done in reverse. That is, the existence of dense Ruzsa–Szemerédi graphs is necessary for efficient equality protocols. This will allow us to deduce both conditional and unconditional communication lower bounds.

▶ **Lemma 24.** *For any s, if there exists a 3-round, s-clean catalytic protocol for equality on n-bit inputs with c bits of catalytic space, there exists an $(\Omega(2^{c-s}),\Omega(2^{n-s}))$-Ruzsa–Szemerédi graph on $O(2^{c+4^s})$ vertices.*

For the purposes of proving this lemma, it would simplify matters to know that Alice only ever modifies the catalytic portion of the blackboard, and Bob only ever modifies the clean portion[6]. As in Section 3, although we cannot claim such behaviour without loss of generality, we can at least claim that, a substantial fraction of the time, Bob only remembers s bits.

▶ **Lemma 25.** *For any $s > 0$, any function $f : \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$ with $\mathsf{CC}_{3,s}(f) = c$ has a protocol of the following form, which we'll call a **sometimes-one-way-catalytic protocol**:*

1. *For every $x \in \{0,1\}^{n_a}$, Alice has an injective function $\alpha^{(x)} : \{0,1\}^c \to \{0,1\}^c \times \{0,1\}^{s\cdot(2^s+1)}$.*
2. *For every $y \in \{0,1\}^{n_b}$, Bob has two functions, $\beta_{rem}^{(y)} : \{0,1\}^c \times \{0,1\}^{s\cdot(2^s+1)} \to \{0,1\}^s$ and $\beta_{out}^{(y)} : \{0,1\}^c \times \{0,1\}^s \to \{0,1\}$.*
3. *Call a pair $(x,\tau) \in \{0,1\}^{n_a} \times \{0,1\}^c$ **bad** if, for some $y \in \{0,1\}^{n_b}$, we have $\beta_{out}^{(y)}(\tau,\beta_{rem}^{(y)}(\alpha^{(x)}(\tau))) \neq f(x,y)$. Then, at most a $2^s/(2^s+1)$ fraction of all pairs are bad.*

[6] In fact, in this case we note that an analysis similar to the one we present here would give a *full*-Ruzsa–Szemerédi graph, as opposed to just a Ruzsa–Szemerédi graph, demonstrating (up to some quantitative loss) an equivalence between full Ruzsa–Szemerédi graphs and this kind of "well-behaved" 3-round catalytic protocol for equality. We suspect that such an equivalence should also exist without the well-behavedness condition – perhaps even that full Ruzsa–Szemerédi graphs can be generically constructed from Ruzsa–Szemerédi graphs – however we do not know a proof.

Proof of Lemma 24. By Lemma 25, we have a sometimes-one-way catalytic protocol for f with c bits of catalytic space. From this protocol, we will construct a bipartite graph G whose 2^c many left vertices are identified with $\{0,1\}^c$, and whose $2^{c+s\cdot(2^s+1)}$ many right vertices are identifed with $\{0,1\}^c \times \{0,1\}^{s\cdot(2^s+1)}$. The edge set will consist of, for every non-bad $(x,\tau) \in \{0,1\}^n \times \{0,1\}^c$, an edge from τ on the left to $\alpha^{(x)}(\tau)$ on the right.

Now, for every $z \in \{0,1\}^n$, $\omega \in \{0,1\}^s$, we'll define a vertex subset $M_z^{(\omega)} \subseteq V(G)$. A left vertex $\tau \in \{0,1\}^c$ will be included in $M_z^{(\omega)}$ if and only if both (z,τ) is non-bad, and $\beta_{\text{rem}}^{(z)}(\alpha^{(z)}(\tau)) = \omega$. A right vertex $\gamma \in \{0,1\}^c \times \{0,1\}^{s\cdot(2^s+1)}$ will be included in $M_z^{(\omega)}$ if and only if $\beta_{\text{rem}}^{(z)}(\gamma) = \omega$.

We claim that every edge of G is induced by one of these subsets, and that each subset induces a matching. The first part of this is immediate: both endpoints of the edge generated by a pair (x,τ) must belong to $M_x^{(\beta_{\text{rem}}^{(x)}(\alpha^{(x)}(\tau)))}$. To see the second part, consider some edge $(\tau,\gamma) \in E(G)$, generated by a non-bad (x,τ), and suppose it belongs to some M_z^{ω} with $z \neq x$. (Note that, by definition of $M_x^{(\omega)}$, this edge it cannot belong to $M_x^{(\omega)}$ for any $\omega \neq \beta_{\text{rem}}^{(x)}(\alpha^{(x)}(\tau))$.) But now, suppose the protocol is run with Alice given input x, Bob given input z, and the catalytic tape initialized to τ. The output of the protocol will be $\beta_{\text{out}}^{(z)}(\tau, \beta_{\text{rem}}^{(z)}(\alpha^{(x)}(\tau))) = \beta_{\text{out}}^{(z)}(\tau, \beta_{\text{rem}}^{(z)}(\gamma)) = \beta_{\text{out}}^{(z)}(\tau,\omega) = \beta_{\text{out}}^{(z)}(\tau, \beta_{\text{rem}}^{(z)}(\alpha^{(z)}(\tau)))$. However, since both (x,τ) and (z,τ) are non-bad, we know $\beta_{\text{out}}^{(z)}(\tau, \beta_{\text{rem}}^{(z)}(\alpha^{(x)}(\tau))) = \mathsf{EQ}_n(x,z) = 0$ and $\beta_{\text{out}}^{(z)}(\tau, \beta_{\text{rem}}^{(z)}(\alpha^{(z)}(\tau))) = \mathsf{EQ}_n(z,z) = 1$, so this is contradiction.

Now, oberve that since there are at least $(2^c \cdot 2^n)/(2^s+1)$ many non-bad pairs, G has at least this many edges. Since we've partitioned these edges into $2^n \cdot 2^s$ many induced matchings, each of which has at most 2^c many edges, we must have at least $2^n/2^{s+1}$ many induced matchings each with at least $2^c/2^{s+1}$ many edges. Letting $G' \subseteq G$ be the graph on only those edges, this is a $2^c + 2^{c+s\cdot(2^s+1)} = O(2^c)$-vertex $(\Omega(2^c), \Omega(2^n))$-Ruzsa–Szemerédi graph. ◀

This characterization immediately allows us to show that no 3-round $\Theta(1)$-clean catalytic protocol for equality can use only $n + O(1)$ bits of catalytic space – although we have found a surprisingly efficient protocol, equality is at least somewhat more difficult for this model than, for instance, inner product.

▶ **Corollary 26.** *For any constant s, we have $\mathsf{CC}_{3,s}(\mathsf{EQ}_n) \geq n + \Omega(\log^*(n))$.*

Proof. It is known by the triangle removal lemma that for any constant k, a $(N/k, t)$-Ruzsa–Szemerédi graph on N vertices must satisfy $t \leq \frac{N}{2^{\Omega(\log^*(N))}}$ [24]. Since, by Lemma 24, a 3-round s-clean protocol on n-bit inputs with c bits of catalytic space gives a $(\Omega(2^c), \Omega(2^n))$-Ruzsa–Szemerédi graph on $O(2^c)$ vertices, this means that $2^n \leq O\left(\frac{2^c}{2^{\Omega(\log^*(2^c))}}\right)$. Rearranging gives $c \geq n + \Omega(\log^* n)$. ◀

Of course, this doesn't rule out a protocol with linear catalytic space – and in fact, the second author believes that such a protocol likely exists. However, it does provide evidence that constructing such a protocol may be difficult: doing so would improve known Ruzsa–Szemerédi constructions, yielding improvements to state-of-the-art bounds in property testing, streaming algorithms, and information theory. We mention a few such implications; the reader is referred to the cited works for definitions of terms.

▶ **Corollary 27.** *If, for some constant s, $\mathsf{CC}_{3,s}(\mathsf{EQ}_n) \leq O(n)$, then*

1. *Any non-adaptive tester for monotonicity over general N-element posets requires query complexity $\Omega(N^c)$ for some $c > 0$.*

2. *Any randomized semi-streaming algorithm for* $(1-\varepsilon)$*-approximate maximum bipartite matching requires* $\Omega(\log(1/\varepsilon))$ *passes over the stream.*
3. *There exist constant-rate centralized coding caching schemes, in which each file is divided into a number of pieces polynomial in the number of participants.*

Proof. By Lemma 24, if $\mathsf{CC}_{3,s}(\mathsf{EQ}_n) \leq O(n)$, then for some $c > 0$ there exist $(\Theta(N), N^c)$-Ruzsa–Szemerédi graphs on N vertices. The property testing lower bounds of Fischer, Lehman, Newman, Raskhodnikova, Rubinfeld and Samorodnitsky [23], the streaming lower bounds of Assadi and Sundaresan [3], and the centralized coding caching scheme construction of Shangguan, Zhang and Ge [44] all rely on constructions of Ruzsa–Szemerédi graphs where the partial matchings are of linear size. Plugging in a construction with $t = N^c$ to those arguments would immediately yield the strengthened bounds in the statement of the corollary. ◀

A more optimistic interpretation of Corollary 27 is that designing catalytic protocols for equality could provide an approach towards improved Ruzsa-Semerédi constructions. There is some evidence that the language of communication complexity can be useful in this area: Linial, Pitassi, and Shraibman have shown a close relationship between Ruzsa–Szemerédi graphs and protocols for high-dimensional permutations in the Number On the Forehead model, which Alon and Shraibman used to give simple communication theoretic descriptions of a couple of known Ruzsa–Szemerédi constructions [35, 1]. 3-round catalytic protocols for equality offer another distinct communication-theoretic interpretation of the problem that may prove useful.

5 The 3-Round Complexity of Indexing

The fact that $\mathsf{CC}_{3,2}(\mathsf{EQ}_n)$ is exponentially smaller than $\mathsf{CC}_{3,1}$ raises the question of whether perhaps *every* function has an efficient 3-round s-clean protocol for some constant s. One might suspect that most functions should require exponential complexity, but there is no clear counting argument to this effect. In fact, even for functions of unbalanced input lengths – i.e. when $n_b \gg n_a$ – there's no obvious way to rule out a protocol with catalytic space close to n_a [7]. We propose considering the following function:

▶ **Definition 13.** *For any* n*, we denote by* IND_n *the function taking an* n*-bit index and a* 2^n*-bit bitstring to the value of that bitstring at that index. That is,*

$$\mathsf{IND}_n : [2^n] \times \{0,1\}^{2^n} \to \{0,1\},$$

$$\mathsf{IND}_n(x, y) = y[x].$$

This indexing function is often considered in one-way communication complexity contexts, where the party holding the index cannot send messages, so all information must be sent by the party with the longer input. Note that in our case, however, we consider the reverse: we're giving Alice the index, and by Lemma 25 our protocol can be largely thought of as one-way from Alice to Bob. The reason we're particularly interested in this setting is that a protocol for IND_n can be thought of as a protocol for all functions simultaneously: Bob has to be able to determine any Boolean function on Alice's input. So, for fixed value of Alice's input length $n_a = n$, IND_n is the hardest possible function, in the sense that a protocol for IND_n gives a protocol with the same parameters for every other function with the same n_a.

[7] The other direction of asymmetry – that is, when $n_a \gg n_b$ – is less interesting for 3-round protocols, since Bob can without loss of generality communicate very little information to Alice, and so there are easily functions requiring at least n_a bits of catalytic space for information theoretic reasons.

By Proposition 4, we know $\mathsf{CC}_{3,s}(\mathsf{IND}_n) \leq 2^n$. We suspect that this is close to the correct answer for any constant s, but the results we've shown thus far haven't ruled out the possibility that $\mathsf{CC}_{3,2}(\mathsf{IND}_n) = n + O(\log^*(n))$. In this section, we will prove a somewhat stronger lower bound – still far away from the upper bound, but enough to suggest for instance that it's unlikely that some sort of Ruzsa–Szemerédi construction as in Section 4 will work directly. We show the following:

▶ **Theorem 9.** *We have $\mathsf{CC}_{3,s}(\mathsf{IND}_n) \geq (1+\varepsilon)n$, for some constant ε depending on s.*

As in our lower bounds for equality, our proof comes from a graph theoretic interpretation. We will show that a protocol for IND_n corresponds to a graph composed of a union of matchings, with the property that any subset of the matchings are separable from the rest of the graph. We will then show that such a graph cannot have too high density, bounding the efficiency of the catalytic protocol. We make the following definitions:

▶ **Definition 28.** *Let E_1 and E_2 be disjoint edge sets on a common set of vertices, and let $k \in \mathbb{N}$. We say E_1 and E_2 are **k-separable** if there exists a k-edge colouring of $E_1 \cup E_2$ such that*

- *each colour appears in only one of E_1 or E_2, and*
- *for every 3-edge path, if the first and last edges share the same colour, so does the middle edge.*

▶ **Definition 29.** *We call an n-vertex graph **k-divisive** if its edges can be partitioned into matchings $M_1, \ldots, M_m$ such that for all $S \subseteq [m]$, the edge sets $\bigcup_{i\in S} M_i$ and $\bigcup_{j\notin S} M_j$ are k-separable.*

▶ **Lemma 30.** *For any constant s, there is a family $\{G_n\}_{n\in\mathbb{N}}$, with $|V(G_n)| \leq O(2^c)$ and $|E(G_n)| \geq \Omega(2^{c+n})$ for $c = \mathsf{CC}_{3,s}(\mathsf{IND}_n)$, and where each G_n is 2^{2^s+s}-divisive.*

Proof. As in Section 4.3, we begin by invoking Lemma 25 to obtain a sometimes-one-way-catalytic protocol. In fact, we'll define the same graph: let G consist of 2^c many left vertices identified with $\{0,1\}^c$, $2^{c+s\cdot(2^s+1)}$ many right vertices identified with $\{0,1\}^c \times \{0,1\}^{s\cdot(2^s+1)}$, and an edge $(\tau, \alpha^{(x)}(\tau))$ for each x whenever (x,τ) is non-bad. This graph has $2^c + 2^{c+s\cdot(2^s+1)} \leq O(2^c)$ many vertices, and $2^{n+c}/(2^s+1) \geq \Omega(2^{c+n})$ many edges.

For any $x \in [2^n]$, let $M_x \subseteq E(G_n)$, $M_x = \{(\tau, \alpha^{(x)}(\tau))\colon (x,\tau) \text{ is not bad}\}$. Note that each M_x is a matching, since $\alpha^{(x)}$ is injective. We claim that, for any $S \subseteq [2^n]$, the edge sets $\bigcup_{i\in S}$ and $\bigcup_{i\notin S} M_i$ are $\left(2^{2^s+s}\right)$-separable. Fix any $S \subseteq [2^n]$, and let $1_S \in \{0,1\}^{2^n}$ denote the indicator vector of S. Now, to separate $\bigcup_{i\in S}$ and $\bigcup_{i\notin S} M_i$, we will "colour" each edge (τ,γ) with the pair $\left(\beta^{(1_S)}_{\text{out}}(\tau,\cdot), \beta^{(1_S)}_{\text{rem}}(\gamma)\right) \in (\{0,1\}^s \to \{0,1\}) \times \{0,1\}^s$. If (τ,γ) and (τ',γ') have the same colour, it means that $\beta^{(1_S)}_{\text{out}}(\tau,\cdot)$ and $\beta^{(1_S)}_{\text{out}}(\tau',\cdot)$ are the same function, and $\beta^{(1_S)}_{\text{rem}}(\gamma)$ and $\beta^{(1_S)}_{\text{rem}}(\gamma')$ are the same bitstring – so, (τ,γ') and (τ',γ) will also share this colour.

We now just need to show that no edge $(\tau,\gamma) \in \bigcup_{i\in S} M_i$ can share a colour with an edge $(\tau',\gamma') \in \bigcup_{i\notin S} M_i$. Fix any $x, x' \in [2^n]$, and any edges $(\tau,\gamma) \in M_x$ and $(\tau',\gamma') \in M_{x'}$ with the same colour. In order for these edges to be present, both (x,τ) and (x',τ') must be good. So,

$$\begin{aligned} f(x,1_S) &= \beta^{(1_S)}_{\text{out}}(\tau, \beta^{(1_S)}_{\text{rem}}(\alpha^{(x)}(\tau))) \\ &= \beta^{(1_S)}_{\text{out}}(\tau, \beta^{(1_S)}_{\text{rem}}(\gamma)) \\ &= \beta^{(1_S)}_{\text{out}}(\tau', \beta^{(1_S)}_{\text{rem}}(\gamma')) \\ &= \beta^{(1_S)}_{\text{out}}(\tau', \beta^{(1_S)}_{\text{rem}}(\alpha^{(x')}(\tau'))) \\ &= f(x',1_S), \end{aligned}$$

meaning that either both x and x' belong to S, or neither do. ◀

We now demonstrate that any such graph cannot contain a large complete bipartite subgraph, which will allow us to bound its density by the KST theorem.

▶ **Lemma 31.** *For any $k \in \mathbb{N}$, if a graph G is k-divisive, then G does not contain the complete bipartite graph $K_{3k^2,3k^2}$ as a subgraph.*

Proof. We proceed by contradiction: suppose we've found a copy H of $K_{3k^2,3k^2}$ in G. Let $M_1 \sqcup \cdots \sqcup M_m = E(G)$ be the partition into matchings guaranteed by the divisiveness condition. We claim that we can find some set of M_i and some smaller complete bipartite subgraph of H whose intersection with the union of those M_i is a perfect matching.

▷ Claim 32. There exists a set $S \subseteq [m]$, and a subgraph $H' \subseteq H \subseteq G$, such that H' is isomorphic to $K_{k+1,k+1}$, and $\bigcup_{i\in S}(H \cap M_i)$ is a matching with $k+1$ edges.

Proof of Claim 32. First, suppose some single M_i contains at least $k+1$ edges of H. Then, the claim follows immediately by taking $S = \{i\}$, and letting H' be the subgraph induced by the edges of $H \cap M_i$. So, we will assume each M_i contains at most k edges of H.

We construct S greedily, starting with $S = \emptyset$ and adding indices one-at-a-time. Suppose $\bigcup_{i\in S}(H \cap M_i)$ currently consists of a matching with $\ell < k+1$ many edges. The number of edges of H that contain any endpoint of an edge in $\bigcup_{i\in S}(H \cap M_i)$ is therefore no more than $2 \cdot \ell \cdot (3k^2)$. Since no M_i contains more than k edges of H, and H has $(3k^2)^2$ edges, there must be at least $\frac{(3k^2)^2}{k} = 9k^3 > 2 \cdot \ell \cdot (3k^2)$ distinct M_i with at least one edge in H. Thus, we can find some i^* such that $M_{i^*} \cap H$ contains at least one edge, but contains no edge sharing an endpoint with an edge of $\bigcup_{i\in S}(H \cap M_i)$ – this implies that $\bigcup_{i\in(S\cup\{i^*\})}(H \cap M_i)$ is a matching with at least $\ell + 1$ many edges. The claim follows by repeating this procedure until $|\bigcup_{i\in S}(H \cap M_i)| \geq k+1$, and then taking H' to be the subgraph induced by those edges. ◁

Now, we use H' and S to contradict the k-separability assumption. By assumption on G, the edge sets $\bigcup_{i\in S} M_i$ and $\bigcup_{i\notin S} M_i$ are k-separable – fix an edge colouring that k-separates them. Since $\bigcup_{i\in S}(H \cap M_i)$ contains $k+1$ edges, by pigeonhole principle it must contain two edges (u_1, u_2) and (v_1, v_2) that are given the same colour. Since H' is a complete bipartite graph, the edge (u_1, v_2) must also be present, and since $\bigcup_{i\in S}(H \cap M_i)$ is a matching it cannot belong to any M_i, $i \in S$. Since (u_1, u_2) and (v_1, v_2) have the same colour, the definition of a k-separation requires (u_1, v_2) to share that colour – but (u_1, v_2) does not belong to $\bigcup_{i\in S} M_i$, so that colour cannot be used on (u_1, v_2). This is contradiction. ◀

Theorem 9 now follows directly from Lemma 30 and Lemma 31.

Proof of Theorem 9. Lemma 30 gives a family $\{G_n\}_{n\in\mathbb{N}}$ of 2^{2^s+s}-divisive graphs, where each G_n has $N = O(2^{\mathsf{CC}_{3,s}(\mathsf{IND}_n)})$ many vertices and $\Omega(2^{\mathsf{CC}_{3,s}(\mathsf{IND}_n)+n}) = \Omega(N^{1+n/\mathsf{CC}_{3,s}(\mathsf{IND}_n)})$ many edges. Then, Lemma 31 ensures that no G_n in this family can contain a copy of $K_{t,t}$, for $t = 3\left(2^{2^s+s}\right)^2$. The Kővári–Sós–Turán theorem guarantees that an N-vertex graph without $K_{t,t}$ as a subgraph must have at most $O(N^{2-1/t})$ many edges [34, 45], so we must have $n/\mathsf{CC}_{3,s}(\mathsf{IND}_n) \leq 1 - 1/t$ for sufficiently large n. Rearranging, this gives that $\mathsf{CC}_{3,s}(\mathsf{IND}_n) \geq n + \left(\frac{1}{t-1}\right) n$ for all sufficiently large n. ◀

This is not an especially strong lower bound – it seems plausible that $\mathsf{CC}_{3,s}(\mathsf{IND}_n)$ is exponential in n for any constant s, but do not even know that $\mathsf{CC}_{3,s}(\mathsf{IND}_n) \geq 2n$. However, the proof does at least give some evidence that the sort of direct application of Ruzsa–Szemerédi constructions we saw in Section 4 is unlikely to work here – Ruzsa–Szemerédi

graphs can contain complete bipartite graphs of any constant size, and we expect that bounds of the form $t \leq n^{1-\Omega(1)}$ for $r = \Theta(n)$ on (r,t)-Ruzsa–Szemerédi graphs, if true, will be difficult to prove [25]. We note also that the graph theoretic property of k-divisiveness, in which any subset of matchings must be k-separable from the rest of the graph, is a strictly stronger property than that of being a Ruzsa–Szemerédi graph, and may be interesting to study in its own right.

6 More Rounds and Connections to Catalytic Computing

We've restricted attention thus far to protocols of only 3 rounds, as the design and analysis of such protocols has proven to already be quite theoretically rich. In this section, we consider what can be said for protocols with more rounds. We observe a close relationship between standard models of nonuniform catalytic computation and our model of catalytic communication, and show that by a communication analogue of Buhrman, Cleve, Koucký, Loff and Speelman's results on CL [13], there exist constant-round constant-clean protocols with polynomial catalytic space for all of TC^0.

6.1 Amortized Bipartite Branching Programs

The notion of catalytic space introduced by Buhrman, Cleve, Koucký, Loff and Speelman is a uniform model of computation: it consists of a space-bounded algorithm, with an additional resource of catalytic space that must be reset at the end of computation [13]. One can also define a natural *nonuniform* catalytic model:

▶ **Definition 33.** *An **amortized branching program**[8] of amortized width w and length ℓ computing m copies of a function f is a directed acyclic multigraph consisting of ℓ layers, where*

- *The first layer has m vertices, the last layer has 2 vertices, and all other layers have mw vertices.*
- *Each layer except the last is labeled with an index $i \in [n]$.*
- *Each vertex except in the last layer has exactly two outgoing edges, labeled 0 and 1 respectively.*
- *The two vertices in the last layer are labeled "accept" and "reject", and the vertices in the second-to-last layer are labeled with the names of vertices in the first layer, such that each name appears exactly w times.*

To compute the function on an input x, the program starts at an arbitrary vertex in the first layer, and then for each layer reads the associated index of x and follows the edge labeled with the resulting value. Correctness of the program entails that, for every input and every starting vertex, the program ends in the vertex with the correct acceptance behaviour, and the second-to-last vertex visited has the same label as the starting vertex.

Amortized branching programs were introduced by Girard, Koucký and McKenzie, who showed examples where direct sum theorems succeeded and failed for the model [26]. Since then, there has been interest in determining the minimum amount of amortization needed to compute a function with small amortized width and length. Potechin showed that every

[8] We remark that our definition is slightly nonstandard, as the last layer is usually defined to have width $2m$ with labels (v,b), where $v \in [m]$ and $b \in \{0,1\}$. However, the definitions are easily seen to be equivalent (up to a unit change in length).

function has an amortized branching program of length $O(n)$ and amortized width $O(1)$ computing 2^{2^n} copies [38]. Robere and Zuiddam showed that this amount of amortization could be reduced for bounded-degree functions over $\mathbb{F}_2$ [42]. Cook and Mertz showed a trade-off in both of these results between length and amortization, allowing them to show in the former case that any function has an amortized branching program of length $O(n)$ and amortized width $O(1)$ computing $2^{2^{\varepsilon n}}$ copies, where the constant in the program length depends on ε [20]. In a recent breakthrough work on the tree evaluation problem, Cook and Mertz showed that the ideas of this tradeoff could be strengthened to give length $\mathsf{poly}(n)$, amortized width $O(1)$ branching programs computing $2^{O(n)}$ copies of any function [21].

In their work on memoryless communication complexity, Arunachalam and Podder observed that the model was closely related to a notion of **bipartite** branching programs [2]. We define a similar notion in the amortized sense, which we note exactly describes our model of catalytic communication, and can be seen as an alternate definition.

▶ **Definition 34.** *A **bipartite amortized branching program** of amortized width w and length ℓ computing m copies of a function f is a directed acyclic multigraph consisting of ℓ layers, where*

- *The first layer has m vertices, the last layer has 2 vertices, and all other layers have mw vertices.*
- *Each layer except the last is labeled either x or y.*
- *Each vertex except in the last layer has exactly 2^n outgoing edges, labeled with the values of $\{0,1\}^n$.*
- *The two vertices in the last layer are labeled "accept" and "reject", and the vertices in the second-to-last layer are labeled with the names of vertices in the first layer, such that each name appears exactly w times.*

To compute the function on an input (x, y), the program starts at an arbitrary vertex in the first layer, and then for each layer follows the edge labeled with the value of the associated input. Correctness of the program entails that, for every input and every starting vertex, the program ends in the vertex with the correct acceptance behaviour, and the second-to-last vertex visited has the same label as the starting vertex.

The equivalence between such programs and catalytic communication protocols is immediate.

▶ **Proposition 35.** *$\mathsf{CC}_{r,s}(f) \leq c$ if and only if there exists a bipartite amortized branching program of amortized width 2^s and length $r + 2$ computing 2^c copies of f.*

Proof. Given such a branching program, we define a catalytic protocol by thinking of the initial catalytic setting as giving the index of a starting vertex of the program, and then on subsequent steps having the blackboard store the index of the current node within its layer. The fact that vertices in the second-to-last layer are labeled by vertices in the first layer means that, with an appropriate ordering of the indexing, the condition of second-to-last vertex having the same label as the starting vertex corresponds exactly to the catalytic portion of the blackboard being reset after r rounds. The final layer corresponds to the output function.

Given a r-round s-clean catalytic protocol with catalytic space c, we can perform the reverse transformation. Each layer except the first and last contains one vertex for each blackboard state, and the transition functions between pairs of layers are determined by how the active player would update the blackboard (with the transition function to the last layer determined by the output function). ◀

Note that any amortized branching program is also a bipartite amortized branching program, so a consequence of this equivalence is that any result known for amortized branching programs trivially extends to catalytic communication. For instance, the results of [20] and [21] give the following, respectively:

▶ **Corollary 36.** $\mathsf{CC}_{O(n),O(1)}(f) \leq 2^{\varepsilon n}$ *for all* $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, $\varepsilon > 0$.

▶ **Corollary 37.** $\mathsf{CC}_{\mathsf{poly}(n),O(1)}(f) \leq O(n)$ *for all* $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$.

However, there are cases where bipartite amortized branching programs can be much more efficient than amortized branching programs. Note that an amortized branching program of length less than n is uninteresting (it is effectively solving a problem on a smaller input length), whereas we've seen that bipartite amortized branching programs of length 5 can be quite powerful. In the full version, we give a result making use of this power: Buhrman, Cleve, Koucký, Loff and Speelman's CL algorithm for threshold circuit evaluation gives an amortized branching program of length $\mathsf{poly}(n)$ even when the circuit has constant depth, but we note that by appropriate rearranging it can be converted to a bipartite amortized branching program of only constant length, yielding the following:

▶ **Theorem 10.** *Let* $f\colon \{0,1\}^{n_a} \times \{0,1\}^{n_b} \to \{0,1\}$ *be any function computable by a size* S*, depth* d *majority circuit with arbitrary input preprocessing – that is, there exists a depth-*d *circuit* C *composed of* S *many majority gates, and arbitrary functions* g, h*, such that* $f(x,y) = C(g(x), h(y))$*. Then,* $\mathsf{CC}_{4^d,1}(f) \leq \mathsf{poly}(S, 2^d)$.

References

1. Noga Alon and Adi Shraibman. Number on the forehead protocols yielding dense ruzsa–szemerédi graphs and hypergraphs. *Acta Mathematica Hungarica*, 161(2):488–506, 2020.
2. Srinivasan Arunachalam and Supartha Podder. Communication Memento: Memoryless Communication Complexity. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 61:1–61:20, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2021.61.
3. Sepehr Assadi and Janani Sundaresan. Hidden permutations to the rescue: Multi-pass streaming lower bounds for approximate matchings. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 909–932, 2023. doi:10.1109/FOCS57990.2023.00058.
4. Felix A Behrend. On sets of integers which contain no three terms in arithmetical progression. *Proceedings of the National Academy of Sciences*, 32(12):331–332, 1946.
5. Y. Birk, N. Linial, and R. Meshulam. On the uniform-traffic capacity of single-hop interconnections employing shared directional multichannels. *IEEE Trans. Inf. Theor.*, 39(1):186–191, 2006. doi:10.1109/18.179355.
6. Béla Bollobás. *Combinatorics: set systems, hypergraphs, families of vectors, and combinatorial probability*. Cambridge University Press, 1986.
7. Mark Braverman, Sumegha Garg, and David P. Woodruff. The coin problem with applications to data streams. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 318–329. IEEE, 2020. doi:10.1109/FOCS46700.2020.00038.
8. Mark Braverman, Sumegha Garg, and Or Zamir. Tight space complexity of the coin problem. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 1068–1079. IEEE, 2021. doi:10.1109/FOCS52979.2021.00106.

9 Mark Braverman, Anup Rao, Ran Raz, and Amir Yehudayoff. Pseudorandom generators for regular branching programs. *SIAM J. Comput.*, 43(3):973–986, 2014. doi:10.1137/120875673.

10 Joshua Brody and Elad Verbin. The coin problem and pseudorandomness for branching programs. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 30–39. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.10.

11 Joshua E. Brody, Shiteng Chen, Periklis A. Papakonstantinou, Hao Song, and Xiaoming Sun. Space-bounded communication complexity. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, pages 159–172, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2422436.2422456.

12 William G Brown, Pál Erdős, and Vera T Sós. On the existence of triangulated spheres in 3-graphs and related problems. *Periodica Mathematica Hungarica*, 3:221–229, 1973.

13 Harry Buhrman, Richard Cleve, Michal Koucký, Bruno Loff, and Florian Speelman. Computing with a full memory: catalytic space. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 857–866. ACM, 2014. doi:10.1145/2591796.2591874.

14 Harry Buhrman, Serge Fehr, Christian Schaffner, and Florian Speelman. The garden-hose model. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, pages 145–158, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2422436.2422455.

15 Harry Buhrman, Michal Koucký, Bruno Loff, and Florian Speelman. Catalytic space: Non-determinism and hierarchy. *Theory Comput. Syst.*, 2018. doi:10.1007/S00224-017-9784-7.

16 Gil Cohen, Dean Doron, Oren Renard, Ori Sberlo, and Amnon Ta-Shma. Error reduction for weighted prgs against read once branching programs. In Valentine Kabanets, editor, *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPIcs*, pages 22:1–22:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.CCC.2021.22.

17 James Cook, Jiatu Li, Ian Mertz, and Edward Pyne. The structure of catalytic space: Capturing randomness and time via compression. *Electron. Colloquium Comput. Complex.*, TR24-106, 2024. URL: https://eccc.weizmann.ac.il/report/2024/106, arXiv:TR24-106.

18 James Cook and Ian Mertz. Catalytic approaches to the tree evaluation problem. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pages 752–760, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3357713.3384316.

19 James Cook and Ian Mertz. Encodings and the tree evaluation problem. In *Electron. Colloquium Comput. Complex*, page 54, 2021.

20 James Cook and Ian Mertz. Trading time and space in catalytic branching programs. In *Proceedings of the 37th Computational Complexity Conference*, CCC '22, Dagstuhl, DEU, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2022.8.

21 James Cook and Ian Mertz. Tree evaluation is in space o(log n · log log n). *Electronic Coloquium Comput. Complex.*, TR23, 2023. URL: https://eccc.weizmann.ac.il/report/2023/174/.

22 Yfke Dulek. Catalytic space: on reversibility and multiple-access randomness, 2015.

23 Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 474–483, New York, NY, USA, 2002. Association for Computing Machinery. doi:10.1145/509907.509977.

24 Jacob Fox. A new proof of the graph removal lemma. *Annals of Mathematics*, pages 561–579, 2011.

25 Jacob Fox, Hao Huang, and Benny Sudakov. On graphs decomposable into induced matchings of linear sizes. *Bulletin of the London Mathematical Society*, 49(1):45–57, 2017.

26 Vincent Girard, Michal Koucký, and Pierre McKenzie. Nonuniform catalytic space and the direct sum for space. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 138, 2015.

27 Ashish Goel, Michael Kapralov, and Sanjeev Khanna. On the communication and streaming complexity of maximum bipartite matching. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 468–485. SIAM, 2012. doi:10.1137/1.9781611973099.41.

28 Chetan Gupta, Rahul Jain, Vimal Raj Sharma, and Raghunath Tewari. Unambiguous catalytic computation. In *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2019*, volume 150 of *LIPIcs*, pages 16:1–16:13, 2019. doi:10.4230/LIPICS.FSTTCS.2019.16.

29 L.H. Harper. Optimal numberings and isoperimetric problems on graphs. *Journal of Combinatorial Theory*, 1(3):385–393, 1966. doi:10.1016/S0021-9800(66)80059-5.

30 Johan Håstad and Avi Wigderson. Simple analysis of graph tests for linearity and pcp. *Random Structures & Algorithms*, 22(2):139–160, 2003. doi:10.1002/RSA.10068.

31 Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In Frank Thomson Leighton and Michael T. Goodrich, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 356–364. ACM, 1994. doi:10.1145/195058.195190.

32 Michael Kapralov. Space lower bounds for approximating maximum matching in the edge arrival model. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1874–1893. SIAM, 2021. doi:10.1137/1.9781611976465.112.

33 Christian Konrad and Kheeran K Naidu. On two-pass streaming algorithms for maximum bipartite matching. *arXiv preprint*, 2021. arXiv:2107.07841.

34 P Kővári, Vera Sós, and Pál Turán. On a problem of zarankiewicz. In *Colloquium Mathematicum*, volume 3, pages 50–57. Polska Akademia Nauk, 1954.

35 Nati Linial, Toniann Pitassi, and Adi Shraibman. On the Communication Complexity of High-Dimensional Permutations. In Avrim Blum, editor, *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 54:1–54:20, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2019.54.

36 Noam Nisan. Pseudorandom generators for space-bounded computation. *Comb.*, 12(4):449–461, 1992. doi:10.1007/BF01305237.

37 Periklis Papakonstantinou, Dominik Scheder, and Hao Song. Overlays and limited memory communication. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 298–308, 2014. doi:10.1109/CCC.2014.37.

38 Aaron Potechin. A note on amortized branching program complexity. In *Proceedings of the 32nd Computational Complexity Conference*, pages 1–12, 2017. doi:10.4230/LIPICS.CCC.2017.4.

39 Edward Pyne. Derandomizing logspace with a small shared hard drive. In Rahul Santhanam, editor, *39th Computational Complexity Conference, CCC 2024, July 22-25, 2024, Ann Arbor, MI, USA*, volume 300 of *LIPIcs*, pages 4:1–4:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.CCC.2024.4.

40 Edward Pyne and Salil P. Vadhan. Pseudodistributions that beat all pseudorandom generators (extended abstract). In Valentine Kabanets, editor, *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPIcs*, pages 33:1–33:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.CCC.2021.33.

41 Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020.

42 Robert Robere and Jeroen Zuiddam. Amortized circuit complexity, formal complexity measures, and catalytic algorithms. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 759–769. IEEE, 2022. doi:10.1109/FOCS52979.2021.00079.

43 I. Ruzsa and E. Szemer'edi. Triple systems with no six points carrying three triangles. *Combinatorica*, 18, January 1976.
44 Chong Shangguan, Yiwei Zhang, and Gennian Ge. Centralized coded caching schemes: A hypergraph theoretical approach. *IEEE Transactions on Information Theory*, 64(8):5755–5766, 2018. doi:10.1109/TIT.2018.2847679.
45 Yufei Zhao. *Graph Theory and Additive Combinatorics: Exploring Structure and Randomness*. Cambridge University Press, 2023.

A Proof of Results In Introduction

First, catalytic protocols with two or fewer rounds cannot meaningfully make use of the catalytic space.

Proof of Proposition 3. In a one-round protocol, the blackboard is only modified once, so if the catalytic portion is ever changed it will end up in a different state from how it started. Since the catalytic state is an arbitrary string that can't be changed, Alice and Bob could just as well do without it.

In a two-round protocol, Alice and Bob each only get to modify the blackboard once. So, Bob must have a function (independent of x) that lets him recover the original state of the catalytic portion for every message Alice can send. There are 2^{s+c} many possible messages for Alice to send, each of which Bob returns to a single fixed catalytic state – so, by pigeonhole principle, the protocol has some catalytic state that's associated with at most 2^s many Alice messages. This gives a new protocol with no catalytic space: the two parties pretend the catalytic portion started in that state, Alice gives Bob the index of which of the 2^s associated messages she would have wanted to send in the original protocol, and Bob determines what bits he would have sent back in the clean space. ◀

Next, we show that every function has a three-round protocol:

Proof of Proposition 4. First, suppose $n_a \leq n_b$. In this case, the protocol is as follows:

1. Alice flips the xth bit of the catalytic portion.
2. Based on his own input, Bob computes the truth table of $x \mapsto f(x, y)$, which he thinks of as a 2^{n_a}-entry bitvector. He takes the inner product of this truth table and the current catalytic space, and writes the result to the clean space.
3. Alice flips the xth bit of the catalytic portion again, resetting it.
4. Bob once again takes the inner product of his truth table and the catalytic space, and outputs the XOR of the result and the clean bit.

If, on the other hand, $n_a > n_b$, the roles are reversed: Alice XORs in the truth table of $y \mapsto f(x, y)$, and Bob reads off the yth bit. In either case, this can be seen as an application of the inner product protocol of Proposition 5 to compute the inner product of a truth table and the indicator vector of the index on which it's being evaluated. ◀

Finally, we present our protocol for inner product.

Proof of Proposition 5. We will describe a protocol. Let x be Alice's input, y be Bob's input, and τ be the initial state of the catalytic space.

1. For the first round of the protocol, Alice replaces the catalytic portion of the blackboard with $\tau \oplus x$, where $\oplus$ denotes bitwise XOR.
2. For the second round, Bob computes the inner product of the catalytic portion and his input, writing the resulting bit $\langle \tau \oplus x, y \rangle$ to the clean space.

3. For the third round, Alice once again bitwise XORs her input into the catalytic portion, resetting it to $(\tau \oplus x) \oplus x = \tau$.
4. To output the answer, Bob computes the inner product of the new catalytic portion and his input, and adds this to the bit stored in the clean space, resulting in $\langle \tau, y \rangle \oplus \langle \tau \oplus x, y \rangle = \langle x, y \rangle$. ◀

Diversity in Evolutionary Dynamics

Yuval Rabani
The Hebrew University of Jerusalem, Israel

Leonard J. Schulman
California Institute of Technology, Pasadena, CA, USA

Alistair Sinclair
University of California, Berkeley, CA, USA

Abstract

Since this paper is under journal submission, we publish only an extended abstract here. A full version can be found at `arXiv:2406.03938 [q-bio.PE]`.

2012 ACM Subject Classification Applied computing → Population genetics; Theory of computation → Algorithmic game theory

Keywords and phrases Mathematical models of evolution, replicator dynamics, weak selection, genetic diversity, game theory, dynamical systems

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.80

Category Extended Abstract

Related Version *Full Version*: `https://arxiv.org/abs/2406.03938`

Funding *Yuval Rabani*: Supported in part by ISF grants 3565-21 & 389-22 and BSF grant 2018687.
Leonard J. Schulman: Supported in part by NSF grant CCF-2321079.
Alistair Sinclair: Supported in part by NSF grant CCF-2231095.

1 Results

We consider the dynamics imposed by natural selection on the populations of two competing, sexually reproducing, haploid species. In this model genomes acquire time-varying fitnesses as a result of the changing mix of species in the population; this is in contrast to most previous works in this area, which impose external rules for varying fitnesses over time. Previous work on our model [2] showed that, in the special case where each of the two species exhibits just two phenotypes, genetic diversity is maintained at all times. This finding supported the tenet that sexual reproduction is advantageous because it promotes phenotype diversity.

In the present paper we consider the more realistic case where there are more than two phenotypes available to each species. The conclusions about diversity in general turn out to be very different from the two-phenotype case. For simplicity, we consider two species with n and m genes respectively, each of which may take two allele values, 0 and 1. Game-theoretically, we think of two teams, A and B, with $|A| = n$ and $|B| = m$. The players within each team play a coordination game, but the two teams between them play a competitive, zero-sum game. Thus for each choice of actions $x \in \{0,1\}^n$, $y \in \{0,1\}^m$ by the $m+n$ players, all players in team A receive common payoff $u(x,y)$ and all players in team B receive $-u(x,y)$, where u is a $2^n \times 2^m$ payoff matrix. Players within a team are not allowed to coordinate their actions, so the strategies of each team are constrained to be product distributions.

Writing $p = (p_1, \ldots, p_n)$ and $q = (q_1, \ldots, q_m)$ for the (time dependent) mixed strategies of teams A, B respectively, where p_i denotes the probability of player i in team A choosing allele 0 (and q_j similarly for player j in team B), the replicator dynamics under weak selection [1] can be written as

$$\dot{p}_i = p_i\big(\widehat{u}_{i,0}(p,q) - \widehat{u}(p,q)\big)\,; \qquad \dot{q}_j = q_j\big(-\widehat{u}_{j,0}(p,q) + \widehat{u}(p,q)\big)\,, \tag{1}$$

© Yuval Rabani, Leonard J. Schulman, and Alistair Sinclair;
licensed under Creative Commons License CC-BY 4.0

where $\widehat{u}(p,q)$ is the average payoff to team A under the current strategies p, q, and $\widehat{u}_{i,0}(p,q)$ (resp., $\widehat{u}_{j,0}(p,q)$) is this same average conditioned on player i of team A (resp., player j of team B) choosing allele 0.

We consider first the following strong notion of preservation of genetic diversity:

▶ **Property A.** *For any initial populations* $(p(0),q(0)) \in (0,1)^{n+m}$ *(i.e., all genotypes are initially represented),*

$$\exists \varepsilon > 0, t_0 \geq 0 \text{ s.t. } \forall t \geq t_0, \quad H(p(t)) + H(q(t)) \geq \varepsilon, \tag{2}$$

where $H(p) = \sum_i H(p_i)$ *and* $H(p_i) = -p_i \log p_i - (1-p_i)\log(1-p_i)$ *is Shannon entropy. I.e., at all times the two-species system maintains diversity (not necessarily within a particular species).*

Our first result is negative: namely, we show that sexual reproduction does not guarantee the maintenance of diversity at all times:

▶ **Theorem 1.** *There exist replicator dynamics with no weak pure Nash equilibrium for which Property A fails. In fact, inequality* (2) *holds only on a set of initial conditions of measure zero.*

Our concrete example used in the proof of Theorem 1 consists of species with three phenotypes each. This is in sharp contrast to the result of [2], which shows that Property A does hold with only two phenotypes per species.

Our main result is a complementary positive statement, which says that in any non-degenerate example, diversity *is* maintained in the following weaker, "infinitely often" sense.

▶ **Property B.** *For any initial populations* $(p(0),q(0)) \in (0,1)^{n+m}$,

$$\exists \varepsilon > 0 \text{ s.t. } \int_0^\infty \max\{0, H(p(t)) + H(q(t)) - \varepsilon\}\, dt = \infty. \tag{3}$$

I.e., during an infinite span of time the entropy is uniformly bounded away from 0.

We also identify the following slightly weaker version of this property:

▶ **Property C.** *For any initial populations* $(p(0),q(0)) \in (0,1)^{n+m}$,

$$\int_0^\infty (H(p(t)) + H(q(t)))\, dt = \infty. \tag{4}$$

The following theorem summarizes the maintenance of diversity in general replicator dynamics, according to whether or not a pure Nash equilibrium exists:

▶ **Theorem 2.** *The following results hold for general replicator dynamics:*

- **(i)** *If the dynamics has no pure Nash equilibrium, then Property B holds.*
- **(ii)** *If the assumption in (i) is weakened to assume only that the dynamics has no* strict *pure Nash equilibrium, then Property C holds.*
- **(iii)** *If the dynamics has a strict pure Nash equilibrium, then Property C (and therefore also Property B) fails on a set of initial populations* $(p(0),q(0)) \in (0,1)^{n+m}$ *of positive measure.*

In summary, our results refute the supposition that sexual reproduction ensures diversity at all times, but affirm a weaker assertion that extended periods of high diversity are necessarily a recurrent event.

References

1 T Nagylaki. The evolution of multilocus systems under weak selection. *Genetics*, 134(2):627–647, June 1993. doi:10.1093/genetics/134.2.627.

2 Georgios Piliouras and Leonard J. Schulman. Learning dynamics and the co-evolution of competing sexual species. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 59:1–59:3. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICS.ITCS.2018.59.

Online Balanced Allocation of Dynamic Components

Rajmohan Rajaraman
Northeastern University, Boston, MA, USA

Omer Wasim
Northeastern University, Boston, MA, USA

Abstract

We introduce Online Balanced Allocation of Dynamic Components (OBADC), a problem motivated by the practical challenge of dynamic resource allocation for large-scale distributed applications. In OBADC, we need to allocate a dynamic set of at most $k\ell$ vertices (representing processes) in $\ell > 0$ clusters. We consider an over-provisioned setup in which each cluster can hold at most $k(1+\varepsilon)$ vertices, for an arbitrary constant $\varepsilon > 0$. The communication requirements among the vertices are modeled by the notion of a *dynamically changing component*, which is a subset of vertices that need to be co-located in the same cluster. At each time t, a request r_t of one of the following types arrives:

1. *insertion* of a vertex v forming a singleton component $\{v\}$ at unit cost.
2. *merge* of (u, v) requiring that the components containing u and v be merged and co-located thereafter.
3. *deletion* of an existing vertex v at zero cost.

Before serving any request, an algorithm can migrate vertices from one cluster to another, at a unit migration cost per vertex. We seek an online algorithm to minimize the total migration cost incurred for an arbitrary request sequence $\sigma = (r_t)_{t>0}$, while simultaneously minimizing the number of clusters utilized. We analyze competitiveness with respect to an optimal clairvoyant *offline* algorithm with identical (over-provisioned) capacity constraints.

We give an $O(\log k)$-competitive algorithm for OBADC, and a matching lower-bound. The number of clusters utilized by our algorithm is always within a $(2+\varepsilon)$ factor of the minimum. Furthermore, in a resource augmented setting where the optimal offline algorithm is constrained to capacity k per cluster, our algorithm obtains $O(\log k)$ competitiveness and utilizes a number of clusters within $(1+\varepsilon)$ factor of the minimum.

We also consider OBADC in the context of machine-learned predictions, where for each newly inserted vertex v at time t: i) with probability $\eta > 0$, the set of vertices (that exist at time t) in the component of v is revealed and, ii) with probability $1-\eta$, no information is revealed. For OBADC with predictions, we give a $O(1)$-*consistent* and $O(\min(\log \frac{1}{\eta}, \log k))$-*robust* algorithm.

2012 ACM Subject Classification Theory of computation → Online algorithms

Keywords and phrases online algorithms, competitive ratio, algorithms with predictions

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.81

Acknowledgements This work was partially supported by NSF award CCF-2335187.

1 Introduction

Modern large-scale applications are distributed over multiple physical machines or computing clusters in data centers, generating a considerable amount of network traffic. In a typical setting, a cloud service provider provisions computational resources (such as CPU, memory, storage) in the form of virtual machines (VMs) to users who pay a cost proportional to their resource requirements which varies over time. However, operating and allocating these resources in a "demand-aware" manner to minimize cost, and optimize performance and energy utilization while fulfilling service-level agreements (SLAs) for various users is technically challenging. This has garnered a surging interest in resource allocation strategies in cloud computing frameworks in recent years [7, 9, 1, 31].


© Rajmohan Rajaraman and Omer Wasim;
licensed under Creative Commons License CC-BY 4.0



Consider the setting where a massively distributed application attributed to a single cloud user may be spread across multiple VMs, necessitating inter-VM communication. If such VM's are allocated in the same cluster, inter-VM communication consumes little network bandwidth, leading to low *communication cost* and potential energy savings [31, 32]. However if inter-VM communication happens over physically separated clusters, this may compromise quality of service, responsiveness, and incur a large consumption of network resources. Almost all virtualization systems support "live-migration" of VMs between various clusters with the aim of reducing latency, improving resource-allocation and energy savings [12, 16, 17]. However, VM migration must be performed sparingly, as it creates network overhead and may compromise the responsiveness of the VM being migrated.

In addition to inter-process communication patterns that evolve over time, another challenge is to provision resources *dynamically* to users under varying resource requirements. This leads to the removal of existing virtual machines and the insertion of new virtual machines [6]. Thereafter, new communication patterns emerge between newly re-allocated VMs. Such re-allocations also incur a cost for the cloud service provider which is inevitable, in addition to future defragmentation of resources caused by multiple re-allocations over time [30].

Finally, *energy consumption* of data centers due to an ever-increasing need for large-scale cloud computing is well documented [19]. In practice, migration of VMs together with other load-balancing strategies have been exploited to reduce the number of "active" physical machines to save energy [16, 19, 17]. For example, VMs on under-loaded clusters are migrated so that such clusters can save energy by moving to an idle state. Thus, it is highly desirable to ensure that the number of "active" clusters *dynamically scales* with the workload.

In this paper, we address the challenges posed by unknown communication patterns, dynamic resource requirements, and energy consumption by introducing the problem of **Online Balanced Allocation of Dynamic Components** (OBADC). We model the scenario where clusters and VMs are homogeneous, and migration and inter-VM communication costs are uniform across different clusters, while the communication patterns and dynamic allocation requests are unknown.

1.1 Problem Formulation and Main Results

Our problem formulation is inspired by the online balanced graph partitioning (OBGR) problem, introduced by Avin, Bienkowski, Loukas, Pacut and Schmid [2], and considers additional technical aspects including the dynamics of VMs, and a secondary goal of minimizing the number of active clusters. In OBADC, we have a set V of at most $n \leq k\ell$ vertices (representing VMs) at any given point in time, and $\ell > 0$ clusters denoted by $S_1, S_2, ..., S_\ell$, each with capacity k. Initially $V = \emptyset$ and S_i for all $i \in [\ell]$ is empty. We model the communication requirements among the vertices using the notion of a dynamically changing *component*, which is a subset of vertices that need to be co-located in the same cluster. The request sequence $\sigma = \{r_t\}_{t \geq 1}$ is revealed online, such that each request r_t is of the following type:

1. *Insertion of a vertex v*: Insert v (and a corresponding singleton component $\{v\}$) at unit cost, to any cluster which can accommodate it. The cost of insertion models a one-time "startup" cost to initialize a VM in a cluster.
2. *Deletion of a vertex v*: Delete existing vertex v from its component C. The cost of deletion is zero; all vertices in $C \backslash \{v\}$ are still required to be co-located.
3. *Merge vertices u and v* : Merge the components containing u and v into a single component so that vertices in the resulting component are co-located thereafter.

Insertions and deletions of vertices model VM allocation and de-allocation, while merge requests model the scenario when frequently communicating VM's dedicated to a single application or a customer are consolidated for efficiency or energy savings [28].

The request sequence σ induces a unique set of components denoted by $\mathcal{C}$, over time. The migration cost of a single vertex is unit. All vertices of a single component are required to be assigned to the same cluster. Thus, we assume that the size of any component $C \in \mathcal{C}$ is most k at all times, and that there exists a component assignment to clusters such that each cluster is assigned a maximum of k vertices.

Competitive Analysis. We adopt the competitive analysis framework to analyze OBADC algorithms. A deterministic (resp. randomized) online algorithm $\mathcal{A}$ is said to be ρ-competitive (alternatively, ρ is the competitive ratio of $\mathcal{A}$) if for any input sequence σ, the cost (resp. expected cost) $c(\mathcal{A}, \sigma)$ incurred by $\mathcal{A}$ in σ satisfies $c(\mathcal{A}, \sigma) \leq \rho \mathrm{OPT}(\sigma) + \gamma$ where γ is a constant independent of the length of σ and $\mathrm{OPT}(\sigma)$ is the cost of an optimal offline algorithm denoted by OPT, which has complete information about the request sequence σ in advance. Abusing notation slightly, we refer to $\mathrm{OPT}(\sigma)$ as OPT when σ is clear from context.

Resource Competitiveness. To capture the practical objective of utilizing a number of clusters proportional to the total resource requirement (which varies over time), we seek algorithms which are also *competitive* with respect to the *number of clusters* utilized by OPT. Let n_t denote the number of vertices at any time t, and let α_t, β_t denote the number of clusters used to assign all n_t vertices at time t by OPT and an online algorithm $\mathcal{A}$. We say that $\mathcal{A}$ is μ-resource competitive if for all time t, $\beta_t \leq \mu \alpha_t$.

OBADC with Over-Provisioning. For the sake of obtaining polynomial-time online algorithms, we consider an overprovisioned setting, similar to the one considered in a recent work [23]. Here, each cluster C_i for $i \in [\ell]$ has capacity $(1+\varepsilon)k$ for some $\varepsilon \in (0, 1)$. While seemingly similar to a $(1+\varepsilon)$ resource-augmented model, we emphasize that in the over-provisioned setup, both the online algorithm and the optimal offline algorithm OPT are constrained to an *identical* cluster capacity of $(1+\varepsilon)k$ for any cluster C_i. As a result, competitiveness is analyzed on a stricter benchmark, in that the optimal offline and online algorithm have access to the same amount of resources, in strong contrast to resource augmentation where competitiveness is analyzed with respect to an unaugmented offline algorithm (see Chapter 4 of [25]).

Our main result is a $O(\log k)$-competitive algorithm for OBADC in the over-provisioned setting. We show that the competitive ratio of our algorithm is asymptotically optimal by establishing a lower bound. Furthermore, our algorithm is $(2+\varepsilon)$-resource competitive.

▶ **Theorem 1.** *Consider any instance of OBADC with ℓ clusters each of capacity $(1+\varepsilon)\ell$ and arbitrary sequence of insertion, deletion, and merge requests such that at any time the resulting components can be allocated to the ℓ clusters using capacity at most k. There exists an online algorithm that achieves cost at most $O(\log k)$ times the optimal, while utilizing at most $(2+\varepsilon)$ times more clusters than the minimum required at each time step.*

Furthermore, every online algorithm incurs cost $\Omega(\log k)$ times the optimal for the worst-case instance.

In a resource augmented setting where the optimal offline algorithm is constrained to capacity k per cluster, our algorithm obtains $O(\log k)$ competitiveness, while utilizing a number of clusters within $(1+\varepsilon)$ factor of the minimum. The proof of resource-competitiveness is deferred to the full version of the paper.

OBADC with Machine-Learned Predictions. We next investigate OBADC in the context of "algorithms with predictions" which is a growing research area [25, 15] with the over-arching goal of circumventing pessimistic limitations of lower and upper bounds for various problems in online, dynamic and approximation algorithms.

Prediction Model: At any time step t when a vertex v is inserted, with probability $\eta \geq 0$, we also receive a set P_v consisting of all vertices in V_t that lie in the same component as v, where V_t denotes the set of vertices at the beginning of time step t. Note that with probability $1-\eta$, we do not receive any information on the component of v. Furthermore, P_v only consists of vertices in v's component which exist at time t, and contains no information about future vertices in v's component which may arrive after time t.

Our "prediction error" is quantified by the confidence $\eta > 0$ of the predictions, similar to [11] and we analyze the performance of a learning-augmented online algorithm for OBADC in a standard *consistency-robustness* framework [11, 15, 24]. An algorithm $\mathcal{A}$ for OBADC with predictions is said to be α-consistent and β-robust if it is α-competitive when $\eta = 1$, and β-competitive where β is a non-increasing function of η. Roughly speaking if $\eta = 1$, and we have perfect predictions, then one would desire constant-factor competitiveness. On the other hand, as η approaches 0, the performance of the algorithm should gracefully degrade to the competitive ratio obtained by the best-known online algorithm for the problem. Our second result provides a near-tight consistency-robustness tradeoff for OBADC.

▶ **Theorem 2.** *There exists an $O(1)$-consistent and $O(\min(\log \frac{1}{\eta}, \log k))$-robust algorithm for OBADC with predictions in the over-provisioned setting.*

1.2 Technical Overview

In this section, we present a high level overview of our techniques towards obtaining our algorithms. We ignore several intricacies in the following discussion.

1.2.1 Overview of the Algorithms

Our algorithm for OBADC classifies components in terms of their volume (which refers to their size). A component is said to be *small* if it has volume at most Dk where $D = \Omega(\varepsilon)$, and large otherwise. Let $\mathcal{C}_S, \mathcal{C}_L$ denote the set of small and large components maintained at all times. Component volumes are rounded up to the nearest power of $(1 + \frac{\varepsilon}{4})$, such that components whose size is in the interval $[(1 + \frac{\varepsilon}{4})^{i-1}, (1 + \frac{\varepsilon}{4})^{i})$ belong to component class i. Thus, the total number of *small* component classes is $O(\log k)$ while the number of *large* component classes is $O(1)$. Our algorithm *assigns* volume to each component C on a cluster such that the assigned volume is upper bounded by $(1 + \frac{\varepsilon}{4})|C|$. The precise volume assigned by our algorithm is more fine-grained to further distinguish components within the same component class.

The assignment of large components is accomplished by solving an integer linear program (ILP) whose objective function seeks to minimize the number of clusters utilized. We remark that large components are assigned *independently* of small components. Since the number of large component classes is $O(1)$, the ILP can be solved in polynomial time. Invoking a result from [26] allows us to limit the total "change" in large component assignments after component merges (resp. vertex deletions), which lead to increase (resp. decrease) in a component's class. Small components are assigned in a *first-fit* fashion. Our algorithm maintains a "tight" assignment of components throughout time which guarantees $(2+\varepsilon)$-resource competitiveness.

The main challenge arises in maintaining these assignments *dynamically*, while simultaneously minimizing the total migration cost and clusters utilized. To accomplish this, our algorithm maintains several invariants. We give an overview of our algorithm as follows.

Insertion of singleton components are handled by provisioning volume on a cluster, with sufficient residual volume whenever possible. For a merge request (u, v), if u, v are in the same component C, the algorithm does nothing; else distinct components C_i, C_j containing u and v respectively, are merged into a new component C_m where w.l.o.g. $|C_i| \geq |C_j|$.

When C_m is *small* and the volume assigned to C_i is at least $|C_m|$, the smaller component C_j is migrated (unless it is assigned on the same cluster as C_i). Otherwise, if there is sufficient residual volume on the cluster to which C_i is assigned, C_m is assigned volume and vertices in C_j are migrated (if necessary). If the residual volume is insufficient, this implies that at least k vertices are assigned to this cluster. Thereafter, vertices in C_i and C_j are migrated to another cluster with sufficient residual volume. This may lead to a reduction in the assigned volume of C_j's cluster. Whenever the residual volume of a cluster decreases below a certain threshold, our algorithm migrates small components as necessary from other clusters to ensure resource-competitiveness.

On the other hand, if C_m is large such that C_m belongs to a higher component class than C_i, our algorithm solves an ILP; otherwise, vertices in C_j are migrated (if necessary). If the ILP is solved, this may lead to multiple components (both large and small) being reassigned and migrated. We give a greedy procedure to assign large components, which together with the sensitivity analysis result from [26], allows us to bound the total migration cost by $O(f(\varepsilon)k)$. If the request sequence consists of only vertex insertions and component merges (i.e. only monotonic increases in component sizes), we can establish $O(\log k)$-competitiveness by a relatively elegant charging argument (see Section 3).

However, for non-monotonic changes in component sizes a major challenge arises towards ensuring resource-competitiveness, since vertex deletions in one component may necessitate migration of other unrelated components to maintain a *tight* packing. To this end, our algorithm always maintains the property that the assigned volume of any component C is within a $(1 + \frac{\varepsilon}{4})$ factor of $|C|$ under both component merges and vertex deletions. For small components undergoing vertex deletions, volume reassignment can be done after a constant fraction of vertices in C are deleted. This may require migration of small components to C's cluster. On the other hand, if vertex deletions in C lead to a drop in C's component class we solve an ILP to reassign large components. This may be prohibitive in general because repeated vertex insertions and deletions can lead to a cost of $O(f(\varepsilon)k)$ every time an ILP is solved. To circumvent this, our algorithm works with a *relaxed* volume threshold (and component class categorization) when "deciding" whether to re-solve an ILP under vertex deletions. Thus, for example, certain large components may be treated as class i components by our algorithm even when their volume is less than $(1 + \frac{\varepsilon}{4})^{i-1}$. However, once at least $\Omega(\varepsilon^2 k)$ vertices are deleted from such components, our algorithm re-solves the ILP. We show that treating certain components in this manner does not violate resource competitiveness, since the total number of large components which can be assigned to any cluster is $O(\frac{1}{\varepsilon})$.

For the learning-augmented model, we enhance our algorithm of Theorem 1 by executing the merges indicated by the predictions at the time of insertion. That is, on the insertion of a vertex v whose predicted set $P(v)$ is non-empty, our algorithm issues merge requests of the form (v, u) for all $u \in P(v)$. This allows us to inherit several desirable properties of our algorithm of Theorem 1 while minimizing the total migration cost when augmented with predictions. The resulting algorithm also yields a smooth interpolation between our algorithm without predictions and one with full predictions.

We present an overview of our analyses in Section 3 and Section 4 for our $O(\log k)$-competitive algorithm and learning-augmented algorithm, respectively.

1.3 Related Work

Online graph partitioning problems. At a high-level, OBADC is related to previous work on online partitioning of dynamically changing graphs. Related problems include variants of online graph coloring and the OBGR problem. In recent work, Azar et al. [3] introduce an online graph recoloring problem in which edges of a graph arrive online and an algorithm needs to distribute the vertices among ℓ clusters such that for any edge, the two endpoints must be in different clusters. Subsequent work [23] considers a capacitated variant of online recoloring. A key distinction between such coloring problems and OBADC is that while an edge in the former requires the endpoints to be *separated*, an edge in the latter requires the endpoints to be *co-located*, leading to different technical considerations.

OBADC is perhaps most closely related to OBGR [2], in which there is a *static* set V of $n = kl$ vertices and a set of $l > 0$ clusters each holding k vertices initially. An online sequence $\sigma = \{(u_t, v_t)\}_{t>0}$ of communication requests arrives over time. The cost of serving request (u_t, v_t) is 1 if u_t and v_t are in distinct clusters and 0 otherwise. Prior to serving any request, a vertex can be migrated at a cost of $\alpha \in \mathbb{Z}^{>0}$. The cost incurred by an online algorithm is the total communication and migration cost to serve all requests in σ. OBGR has a $O(k^2l^2)$-competitive algorithm [2] in the non-augmented setting while a lower bound of $\Omega(kl)$ is known. In recent work, a $O(kl2^{O(k)})$-competitive algorithm was given in [4], which is optimal for constant k. In the resource augmented setting, with $(2+\varepsilon)$-augmented cluster capacity an $O(k \log k)$-competitive deterministic algorithm was given in [2]. For the same setting, a polynomial time algorithm was recently presented in [10]. A lower bound of $\Omega(l)$ holds even in the $(1+\varepsilon)$-augmented setting where $\varepsilon < 1/3$ [20].

In [14], Henzinger et al. introduced the learning model of the problem in which the vertex set of the graph G_σ induced by the request sequence σ can be partitioned into $V_1, ...V_l$ such that $|V_i| = k$ for all i and there are no inter-cluster requests (u, v) where $u \in V_i.v \in V_j$ for $j \neq i$. For the learning model, the lower bound of $\Omega(kl)$ holds in the non-augmented setting. Allowing $(1+\varepsilon)$-augmentation, the best deterministic and randomized algorithms are $\Theta(l \log k)$ and $\Theta(\log k + \log l)$-competitive respectively which are shown to be tight[13].

OBADC vs. OBGR. We describe the similarities and differences between OBADC and OBGR. OBADC resembles a "fully dynamic" version of OBGR in the learning model, but differs as follows. In OBADC, vertices arrive and depart over time, and hence the online algorithm has to adapt to communication patterns which evolve over time between both existing vertices and new vertices. The migration costs are identical between the two models; however in OBADC, an insertion of a new vertex incurs a cost of 1 and we aim to have the total number of clusters utilized by an online algorithm to adapt with the number of vertices. This may incur strategic migrations of vertices to satisfy this additional objective, in contrast to OBGR where migrations are performed to solely minimize communication overhead.

Resource-augmentation and over-provisioning. To circumvent pessimistic lower-bounds and obtain polyonomial time algorithms, many online algorithmic problems have been studied in the *resource-augmented* model (e.g., see [21, 33, 34]), going back as far as the earliest work on caching [29]. OBGR has also been studied in the resource-augmentation model [13, 22, 10, 2], where OPT is constrained to a capacity of k on each cluster, while an online algorithm can utilize capacity $(1+\varepsilon)k$ for some $\varepsilon > 0$. In contrast, we study OBADC in an *over-provisioned* setting, which yields stronger guarantees than resource-augmentation since the algorithm and the optimal have the same resources. Any competitive ratio in

the over-provisioned model immediately implies the same bound in the resource augmented model; for the measure of number of clusters utilized, we are able to get stronger bounds in the resource-augmentation model (see the remark after Theorem 1 in Section 1.1).

Learning-augmented algorithms. Another approach toward addressing pessimistic bounds in online and approximation algorithms is to incorporate the growing availability of machine-learned predictions in the design of algorithms [11, 18, 15, 5]. A central theme in this area is the following: given access to an imperfect machine-learned oracle which can "predict" characteristics of future request sequences or actions (in the context of online algorithms [24, 15]), edge updates (in the context of dynamic algorithms [5]) or optimal solutions (in the context of approximation algorithms [8]), can improved competitive ratios, running/update times, or approximation ratios be obtained? The imperfection of predictions is quantified by prediction error, which is problem and instance dependent, or a confidence parameter [11]. There is a great variation in terms of various prediction models and prediction errors which have been utilized and there is no singular notion of either, which can be useful or applicable to wide variety of problems.

Our study of online OBADC with predictions is also inspired from prior empirical work in VM allocation and migration, in which various predictions on resource utilization in clusters [17], future VM requests and migrations [19], and communication traffic from historically collected data have been utilized towards efficient provisioning of resources.

1.4 Outline and Preliminaries

Section 2 presents our main algorithm for OBADC. In Section 3, we give a proof sketch of the competitive ratio of our algorithm. The proofs of resource competitiveness and correctness are deferred to the full version of the paper. Finally, Section 4 presents a learning-augmented algorithm for OBADC with predictions, together with a proof sketch of Theorem 2.

We present some useful notation. Let V_t denote the set of vertices at any given point in time t such that $n_t = |V_t| \leq kl$. We use the notation $[i]$ for any $i \in \mathbb{Z}^+$ to refer to the set $\{1, 2, \ldots, i\}$. The volume of a component C is simply its size, and denoted by $|C|$.

Our algorithms maintain a set of components $\mathcal{C} = \{C_1, C_2, ...\}$ induced on V_t for all time t. We use $\mathcal{S} = \{S_1, S_2, \ldots, S_\ell\}$ to denote the set of all clusters. When there is merge a request (u, v) between distinct components C_i and C_j, our algorithms merge C_i and C_j into C_m. A merge is viewed as deletion of components C_i and C_j from $\mathcal{C}$ and an insertion of C_m to $\mathcal{C}$.

2 An $O(\log k)$-competitive algorithm

In this section, we present an *asymptotically* optimal deterministic $O(\log k)$-competitive algorithm for OBADC. We begin by giving some definitions and invariants that our algorithm maintains.

2.1 Definitions and Invariants

Component Classes. Our algorithm classifies components according to their size. For a component $C \in \mathcal{C}$, we say that C is in class i if $|C| \in [(1 + \frac{\varepsilon}{4})^{i-1}, (1 + \frac{\varepsilon}{4})^{i})$, where $1 \leq i \leq \lfloor \ln_{(1+\frac{\varepsilon}{4})} k \rfloor$. We say that a component is small if it is in class i where $i \leq c_s$ where $c_s \leq \lfloor \ln_{(1+\frac{\varepsilon}{4})} \frac{\varepsilon k}{4} \rfloor$. Thus, any small component has size at most Dk where $\frac{\varepsilon}{5} \leq D \leq \frac{\varepsilon}{4}$. All the other components are said to be large.

A large component is understood to be in class i if it is in class $i + c_s$. By the above definitions, it follows that a large component of class i has size in $[Dk(1+\frac{\varepsilon}{4})^{i-1}, Dk(1+\frac{\varepsilon}{4})^{i})$. The following lemma bounds the number of large component classes.

▶ **Lemma 3.** *The number of large component classes is* $c_l \leq \frac{26}{\varepsilon^2} = O(1)$.

Proof. Note that a large component has volume at least $\frac{\varepsilon k}{5}$ and at most k. Thus, $c_l \leq \frac{\ln(\frac{5}{\varepsilon})}{\ln(1+\frac{\varepsilon}{4})} + 1$. Since $1 - \frac{1}{x} \leq \ln x \leq x - 1$ for any $x \in \mathbb{R}^{\geq 0}$, it follows that $c_l \leq \frac{(5/\varepsilon)}{1/(1+\varepsilon/4)} \leq \frac{4+\varepsilon}{\varepsilon} \cdot \frac{5}{\varepsilon} + 1 \leq \frac{26}{\varepsilon^2}$ since $\varepsilon < 1$. ◀

Assigned Volume. For every component $C \in \mathcal{C}$, our algorithm provisions volume on some cluster $S_i \in \mathcal{S}$ to which it is assigned, which we refer to as the assigned volume and denote it by $A(C)$.

For any cluster $S_i \in \mathcal{S}$ let $A(S_i)$ denote the total assigned volume for all components assigned to S_i. Let R_i denote the residual *unassigned* volume of S_i, where $R_i = (1+\varepsilon)k - A(S_i)$. We present a fine-grained scheme to assign volumes to components as follows.

Let $A_{ij} = (1 + j\frac{\varepsilon}{16})(1+\frac{\varepsilon}{4})^{i-1}$, for $j \in \{0, 1, 2, 3, 4\}$. Thus, $A_{i0} = (1+\frac{\varepsilon}{4})^{i-1}$ and $A_{i4} = (1+\frac{\varepsilon}{4})^{i}$. Our algorithm maintains the following invariant for assigned volumes at all times.

▶ **Invariant 1** (Assigned Volume). *For all classes i, and any class i component such that $|C| \in [(1+\frac{\varepsilon}{4})^{i-1}, (1+\frac{\varepsilon}{4})^{i})$, one of the following statements holds:*

1. *if $A_{i0} \leq |C| < A_{i1}$, then $|C| \leq A(C) \leq A_{i3}$.*
2. *if $A_{i1} \leq |C| < A_{i2}$, then $|C| \leq A(C) \leq A_{i4}$.*
3. *if $A_{i2} \leq |C| < A_{i3}$, then $|C| \leq A(C) \leq (1+\frac{\varepsilon}{16})(1+\frac{\varepsilon}{4})^{i} = A_{i+1,1}$.*
4. *if $A_{i3} \leq |C| < A_{i4}$, then $|C| \leq A(C) \leq (1+\frac{\varepsilon}{8})(1+\frac{\varepsilon}{4})^{i} = A_{i+1,2}$.*

The following lemma is immediate based on the invariant above.

▶ **Lemma 4.** *If Invariant 1 is maintained, then for any component C, $|C| \leq A(C) \leq (1+\frac{\varepsilon}{4})|C|$.*

Active Clusters. A cluster is said to be *active* if there is at least one component assigned to it. The set of all active clusters is denoted by $\mathcal{S}_A$.

Marked Clusters. Our algorithm marks and un-marks active clusters over time. If a cluster is marked, then its residual volume is small. We denote the set of marked clusters by $\mathcal{S}_M$, where $\mathcal{S}_M \subseteq \mathcal{S}_A$. Initially, all clusters are unmarked, i.e. $\mathcal{S}_M = \emptyset$. Our algorithm maintains the following invariant at all times.

▶ **Invariant 2.** *For a cluster $S_j \in \mathcal{S}_M$, $R_j \leq \frac{\varepsilon k}{3}$.*

Furthermore, let $\mathcal{S}_S$ denote the set of clusters on which only small components (and no large components) are assigned and let $\mathcal{S}_L$ denote the set of clusters to which at least one large component is assigned. To ensure that the number of active clusters is at most $(2+\varepsilon)$ times the minimum needed, our algorithm maintains the following invariant.

▶ **Invariant 3.** *At any time t, if $|\mathcal{S}_S| > 0$ then both of the following hold: i) either $\mathcal{S}_L = \emptyset$ or $\mathcal{S}_L \subseteq \mathcal{S}_M$ (i.e. all clusters in $\mathcal{S}_L$ are marked) and, ii) exactly one cluster in $\mathcal{S}_S$ is unmarked.*

Marked Components. Our algorithm marks and un-marks certain components when their component class decreases due to vertex deletions. If a large component of class $i \in [c_l - 1]$, (resp. a small component of class c_s) is marked, then it is treated as a large component of class $i+1$ (resp. a large component of class 1). Intuitively, we mark a component to ensure that the ILP is not solved too frequently if its volume shrinks only slightly due to deletions. The set of marked components at any time is denoted by $\mathcal{C}_M$. Note that all marked components are *treated* as large components by our algorithm, thus $\mathcal{C}_M \subseteq \mathcal{C}_L$. Our algorithm maintains the following invariant for marked components.

▶ **Invariant 4.** *At any point in time, a marked component $C \in \mathcal{C}_L$ satisfies one of the following conditions:*

1. *C is a small component of class c_s, such that $|C| \geq Dk - \frac{\varepsilon^2 k}{100}$.*
2. *C is a large component of class $i \in [c_l - 1]$, such that $|C| \geq Dk(1+\frac{\varepsilon}{4})^i - \frac{\varepsilon^2 k}{100}$.*

Data Structures. Our algorithms maintain the sets $\mathcal{C}_S$ and $\mathcal{C}_L$ of small and large components, respectively, together with the set $\mathcal{C}_M$ of marked components. For a component C, the assigned volume $A(C)$ is maintained, and for all $S_i \in \mathcal{S}$, $A(S_i)$ and R_i are maintained. Furthermore, the sets of clusters $\mathcal{S}_A, \mathcal{S}_M, \mathcal{S}_S$ and $\mathcal{S}_L$ are maintained.

In the following section, we outline our approach for assigning large components which is completely independent of the assignment of small components.

2.2 An Approach to Assign Large Components

We first define the notion of a signature, similar to [22, 13]. Intuitively, a signature encodes the number of large components of each component class that can be assigned to a single cluster while respecting cluster capacity.

▶ **Definition 5.** *A signature $\tau = (\tau_1, \tau_2, ..., \tau_{c_l})$ is a non-negative vector of dimension c_l where τ_i denotes the number of large components of class i such that $\sum_{i=1}^{c_l}(Dk(1+\frac{\varepsilon}{4})^{i-1} - \frac{\varepsilon^2 k}{100})\tau_i \leq k$.*

Let $\mathcal{T} = \{T_1, T_2, ...\}$ denote the set of all possible signatures. The following lemma bounds the size of $\mathcal{T}$.

▶ **Lemma 6.** *The total number of signatures, $|\mathcal{T}|$ is bounded by $(\frac{6}{\varepsilon})^{\frac{26}{\varepsilon^2}} = O(1)$.*

Proof. We first note that $Dk(1+\frac{\varepsilon}{4})^{i-1} - \frac{\varepsilon^2 k}{100} \geq Dk - \frac{\varepsilon^2 k}{100} \geq \frac{\varepsilon k}{5} - \frac{\varepsilon^2 k}{100} \geq \frac{\varepsilon k}{6}$ for any $i > 0$. Thus, any entry τ_i is upper bounded by $\frac{k}{\varepsilon k/6} = \frac{6}{\varepsilon}$. Since the number of large component classes is $\frac{26}{\varepsilon^2}$ by Lemma 3, the total number of signatures is at most $(\frac{6}{\varepsilon})^{\frac{26}{\varepsilon^2}} = O(1)$. ◀

We solve an integer linear program (ILP) in polynomial time, and obtain signatures which guide the placement of large components. Let T_{ij} denote the j^{th} entry of a signature $T_i \in \mathcal{T}$. For each signature T_i, we let $x_i \in [0, \ell]$ denote the number of clusters which are assigned signature T_i. Let σ_j denote the number of class j large components at any given point in time. Our ILP is as follows.

ILP.

$$\min \quad \sum_{i=1}^{|\mathcal{T}|} x_i \quad \text{s.t.} \quad \sum_{i=1}^{|\mathcal{T}|} T_{ij} x_i = \sigma_j \ \forall j \in [c_l], \quad x_i \in [0, \ell] \ \forall i \in [\mathcal{T}].$$

In matrix form, our ILP has $n_r = O(c_l) = O(1)$ rows and $n_c = O(|\mathcal{T}|) = O(1)$ columns. Thus, the ILP can be solved in polynomial time.

Our algorithm solves this ILP whenever σ_j changes for any $j \in [c_l]$. This could be after components are merged leading to a larger component with a higher component class or, vertex deletions from a component leading to a decrease in its component class. In both cases, at most three components in $\mathcal{C}$ change, and as a result, at most three component classes change. We invoke a well-known result in sensitivity analysis from [27], which quantifies the *change* in optimal solutions of the ILP.

▶ **Theorem 7** ([27]). *Let A be an integral $n_r \times n_c$ matrix, such that each subdeterminant of A is at most Δ in absolute value; let b' and b'' be column n_r-vectors, and let c be a row n_c-vector. Suppose $\max\{cx|Ax \leq b' : x \text{ integral}\}$ and $\max\{cx|Ax \leq b'' : x \text{ integral}\}$ are finite. Then for each optimum solution z' of the first maximum there exists an optimum solution z'' of the second maximum such that $\|z' - z''\|_\infty \leq n_c\Delta(\|b' - b''\|_\infty + 2)$.*

We prove the following lemma.

▶ **Lemma 8.** *After a merge request or deletion request, the number of signatures which change in the optimal solution to the ILP is $f(\varepsilon) = \frac{|\mathcal{T}|^{2+|\mathcal{T}|}}{\varepsilon^{|\mathcal{T}|}} = O(1)$ for constant $\varepsilon > 0$.*

Proof. Whenever two components are merged or a deletion causes a large component's size class to decrease, the RHS vector in our ILP changes by at most 1 in the infinity norm. To bound the sub-determinant, we use the Hadamard inequality to derive that $\Delta \leq (n_c A_{\max})^{n_c/2}$ where $A_{\max}$ denotes the maximum entry (in absolute value) of the constraint matrix A. Each entry in the constraint matrix of our ILP has value either 1 or T_{ij} so that $A_{\max} \leq \frac{k}{Dk} = O(\frac{1}{\varepsilon})$. As a result, $\Delta = O((\frac{|\mathcal{T}|}{\varepsilon})^{|\mathcal{T}|})$. Thus, the optimal solution to the ILP changes by $O(|\mathcal{T}|\Delta)$ in the infinity norm. Since x has dimension $|\mathcal{T}|$, the number of signatures which change between any two optimal solutions is $O(|\mathcal{T}|^2\Delta) = O(1)$. ◀

Greedy Assignment of Signatures to Clusters. Given the solution $x = (x_1, x_2, ..., x_{|\mathcal{T}|})$ from the ILP, we perform a *greedy* assignment of signatures to clusters which minimizes the number of clusters whose assigned signatures change. Let $x' = (x'_1, x'_2, ..., x'_{|\mathcal{T}|})$ denote the current signature, such that there are exactly x'_i clusters which are assigned signature T_i. Our algorithm for greedy assignment, `Assign-Signatures` works as follows.

Let $z_i = x_i$ for all $i \in [|\mathcal{T}|]$ and $\mathcal{S}' := \mathcal{S}_A$ denote the set of active clusters. We iterate over all $i \in [|\mathcal{T}|]$: while $z_i > 0$, if there exists an active cluster S_j in $\mathcal{S}'$ which is currently assigned signature T_i, z_i is decremented, S_j is removed from $\mathcal{S}'$ and we continue. Else if no cluster in $\mathcal{S}'$ is assigned signature T_i, we pick an arbitrary active cluster S_j in $\mathcal{S}_A$ and assigns T_i to S. If $\mathcal{S}' = \emptyset$, our algorithm picks an arbitrary inactive cluster $S_j \in \mathcal{S} \backslash \mathcal{S}_A$, adds S_j to $\mathcal{S}_A$ and assigns T_i to S.

Let $\mathcal{S}_N \subseteq \mathcal{S}_A$ denote the set of clusters which are assigned a new signature after the call to `Assign-Signatures`. Any cluster $S \in \mathcal{S}_N$ satisfies one of the following conditions prior to the call to `Assign-Signatures`: i) S was inactive, ii) S was active but its signature changed after the call to `Assign-Signatures` or iii) S was active and was not assigned any signature prior to the call to `Assign-Signatures`.

Let $\mathcal{S}_O \subseteq \mathcal{S}$ denote the set of clusters which were assigned a signature prior to solving the ILP, and are no longer assigned a signature. Finally, let $\mathcal{S}_I$ denote the set of clusters whose assigned signatures are identical before and after the call to `Assign-Signatures`. Our algorithm returns sets $\mathcal{S}_N, \mathcal{S}_O$ and $\mathcal{S}_I$. Clearly, the assigned signatures change only for clusters which are in $\mathcal{S}_N \cup \mathcal{S}_O$.

By virtue of subroutine `Assign-Signatures` and Lemma 8, the following lemma is immediate.

Algorithm 1 `Assign-Signatures`.

Let $x' = (x'_1, x'_2, ..., x'_{|\mathcal{T}|})$ denote the current signature.
$\mathcal{S}' \leftarrow \mathcal{S}_A$, $\mathcal{S}_N \leftarrow \emptyset$, $\mathcal{S}_I \leftarrow \emptyset$.
Solve the ILP and let $x = (x_1, x_2, ..., x_{|\mathcal{T}|})$ denote the obtained signature
for $i \in [|\mathcal{T}|]$ **do**
    $z_i = x_i$.
    while $z_i > 0$ **do**
        if there exists $S_j \in \mathcal{S}'$ assigned a signature T_i **then**
            $z_i \leftarrow z_i - 1$, $\mathcal{S}' \leftarrow \mathcal{S}' \backslash \{S_j\}$, $\mathcal{S}_I \leftarrow \mathcal{S}_I \cup \{S_j\}$.
        else
            if $\mathcal{S}' = \emptyset$ **then**
                Pick an arbitrary cluster $S_j \in \mathcal{S} \backslash \mathcal{S}_A$, and assign T_i to S_j.
                $\mathcal{S}_N \leftarrow \mathcal{S}_N \cup \{S_j\}$.
            else
                Pick an arbitrary active cluster S_j in $\mathcal{S}'$ and assign T_i to S_j.
                $\mathcal{S}_N \leftarrow \mathcal{S}_N \cup \{S_j\}$, $\mathcal{S}' \leftarrow \mathcal{S}' \backslash \{S_j\}$.
$\mathcal{S}_O \leftarrow \mathcal{S}_L \backslash (\mathcal{S}_I \cup \mathcal{S}_N))$.
$\mathcal{S}_L \leftarrow \mathcal{S}_N \cup \mathcal{S}_I$.
Return $(\mathcal{S}_N, \mathcal{S}_O, \mathcal{S}_I)$.

▶ **Lemma 9.** *After a call to* `Assign-Signatures`, $|\mathcal{S}_N \cup \mathcal{S}_O| = f(\varepsilon) = O(1)$.

For all clusters $S_i \in \mathcal{S}_I$ whose assigned signatures do not change, our algorithm does not modify the assignment of large components. Let $\mathcal{C}'_L$ denote the set of all large components in $\mathcal{C}_L$ which are not assigned to any cluster in $\mathcal{S}_I$. It follows from Lemma 9 that the total volume of all large components in $\mathcal{C}'_L$ is $O(kf(\varepsilon))$.

Components in $\mathcal{C}'_L$ are assigned to clusters in $\mathcal{S}_N$, corresponding to the newly assigned signatures. Let $\mathcal{C}'_S$ denote the set of all small components previously assigned to $\mathcal{S}_N \cup \mathcal{S}_O$. From Lemma 9, it follows that the total volume of all components in $\mathcal{C}'_S$ is $O(kf(\varepsilon))$. Our algorithm re-assigns small components in $\mathcal{C}'_S$ in a *first-fit* fashion, explained in the next section. The following Lemma is immediate from Lemma 8, Lemma 9 and our preceding discussion.

▶ **Lemma 10.** *The total volume of components in* $\mathcal{C}'_L \cup \mathcal{C}'_S$ *is bounded by* $O(kf(\varepsilon) = O(k)$.

2.3 The Full Algorithm

In this section, we present our algorithm which takes as input a request r_t at time t, and maintains an assignment of components to clusters while satisfying Invariants 1 through 4. Our algorithm relies on various subroutines, which we present in the following.

Subroutine `Assign-Volume`. This subroutine takes as input a component C, and a cluster $S_j \in \mathcal{S}_A$ and assigns volume for C on S_j. In the case when C is a class $i-1$ marked component, it is treated as a class i component and assigned volume A_{i2}.

Observe that Invariant 1 is satisfied for any component C after a call to `Assign-Volume`.

Subroutine `Assign-Small`. The subroutine `Assign-Small` is invoked whenever a small component C needs to be assigned volume on a cluster. We iterate over active unmarked clusters $S_j \in \mathcal{S}_A \backslash \mathcal{S}_M$, and if $R_j \geq (1 + \frac{\varepsilon}{4})|C|$, C is assigned to S_j. Else, S_j is marked and added to $\mathcal{S}_M$.

Algorithm 2 `Assign-Volume`(C, S_j).

if $C \notin \mathcal{C}_M$ **then**
  Let $i > 0$ s.t. $|C| \in [(1+\frac{\varepsilon}{4})^{i-1}, (1+\frac{\varepsilon}{4})^{i})$.
  $A(C) \leftarrow 0$.
  if $A_{i0} \leq |C| < A_{i1}$ **then**
    $A(C) \leftarrow A_{i2}$.
  else if $A_{i1} \leq |C| < A_{i2}$ **then**
    $A(C) \leftarrow A_{i3}$.
  else if $A_{i2} \leq |C| < A_{i3}$ **then**
    $A(C) \leftarrow A_{i,4}$.
  else
    $A(C) \leftarrow A_{i+1,1}$.
  $A(S_j) \leftarrow A(S_j) + A(C)$, $R_j \leftarrow R_j - A(C)$.
else ▷ If C is marked.
  Let $i \geq 0$ s.t. $|C| \in [Dk(1+\frac{\varepsilon}{4})^{i-1} - \frac{\varepsilon^2 k}{100}, Dk)$.
  $A(C) \leftarrow A_{i2}$.

If all clusters are marked, i.e. $\mathcal{S}_M = \mathcal{S}_A$ we add an arbitrary cluster in $\mathcal{S} \backslash \mathcal{S}_A$ to $\mathcal{S}_A$ and assign volume for C.

Algorithm 3 `Assign-Small`(C).

is-assigned=false.
for all $S_j \in \mathcal{S}_A \backslash \mathcal{S}_M$ **do**
  if $R_j \geq (1+\frac{\varepsilon}{4})|C|$ **then**
    `Assign-Volume`(C, S_j).
    is-assigned=true.
    break
  else
    $\mathcal{S}_M \leftarrow \mathcal{S}_M \cup \{S_j\}$.
if is-assigned=false **then**
  Pick an arbitrary cluster $S_j \in \mathcal{S} \backslash \mathcal{S}_A$.
  $\mathcal{S}_A \leftarrow \mathcal{S}_A \cup \{S_j\}$.
  `Assign-Volume`(C, S_j).
  $\mathcal{S}_S \leftarrow \mathcal{S}_S \cup \{S_j\}$.

Subroutine `Handle-Insertion`. The subroutine `Handle-Insertion` is called whenever a request r_t at time t corresponds to a vertex insertion v. Subroutine `Assign-Small` is invoked to assign the singleton component $\{v\}$.

Algorithm 4 `Handle-Insertion`(v).

$C \leftarrow \{v\}$.
$\mathcal{C}_S \leftarrow \mathcal{C}_S \cup \{C\}, \mathcal{C} \leftarrow \mathcal{C} \cup \{C\}$.
`Assign-Small`(C).

Subroutine `Unassign-Volume`. This subroutine takes as input a component C which is currently assigned to a cluster S_i. The quantity $A(S_i)$ is decremented by $A(C)$ and $A(C)$ is set to 0. If $S_i \in \mathcal{S}_M$ and $R_i \geq \frac{\varepsilon k}{3}$, S_i is unmarked. The reason why S_i is unmarked is that if $R_i \geq \frac{\varepsilon k}{2}$, a small component can be assigned volume on S_i, since the assigned volume for a small component is bounded by $\frac{\varepsilon k}{4}(1+\frac{\varepsilon}{4}) < \frac{\varepsilon k}{3}$ by Invariant 1. The pseudo-code is given as follows.

Algorithm 5 `Unassign-Volume`(C, S_i).

1: $A(S_i) \leftarrow A(S_i) - A(C)$, $R_i \leftarrow R_i + A(C)$.
2: $A(C) \leftarrow 0$.
3: **if** $S_i \in \mathcal{S}_M$ and $R_i \geq \frac{\varepsilon k}{2}$ **then**
4: $\quad \mathcal{S}_M \leftarrow \mathcal{S}_M \backslash \{S_i\}$.

Subroutine `Handle-Unmarked`. This subroutine takes as input a newly unmarked cluster S_i. If the assigned volume $A(S_i) = 0$, then S_i is removed from the sets $\mathcal{S}_A$ and $\mathcal{S}_S$. If $|\mathcal{S}_S| > 0$, an arbitrary cluster $S \in \mathcal{S}_S$ is chosen and unmarked.

Else, if $A(S_i) > 0$, Invariant 3 is restored as follows. Note that if $S_i \in \mathcal{S}_L$ and $\mathcal{S}_S = \emptyset$ or $S_i \in \mathcal{S}_S$ and $|\mathcal{S}_S| = 1$, nothing needs to be done since Invariant 3 is already maintained. Else, let $S \in \mathcal{S}_S$ denote an unmarked cluster where $S \neq S_i$. While the residual volume of S_i is sufficient (and S_i is unmarked and $|\mathcal{S}_S| > 0$) small components from S are migrated to S_i. If the residual volume of S_i is insufficient, S_i is marked and added to $\mathcal{S}_M$. If at any point $A(S) = 0$, S is removed from $\mathcal{S}_A$ and $\mathcal{S}_S$. Thereafter, if $S_i \in \mathcal{S}_L$, and $|\mathcal{S}_S| > 0$, an arbitrary cluster S in $\mathcal{S}_S$ is unmarked. If $S_i \in \mathcal{S}_S$, and $|\mathcal{S}_S| > 1$, an arbitrary cluster in $\mathcal{S}_S$ is unmarked. The procedure continues while the residual volume of S_i is sufficient. On the other hand, if S_i is the only cluster in $\mathcal{S}_S$ it is unmarked and the process terminates.

Subroutine `Reassign-Large`. We give a subroutine `Reassign-Large` which is invoked whenever the number of large components σ_i of any class $i \in [c_l]$ change. In `Reassign-Large`, the subroutine `Assign-Signatures` is invoked which returns the sets $\mathcal{S}_N, \mathcal{S}_O$ and $\mathcal{S}_I$ of clusters (see Section 2.2).

Let $\mathcal{C}'_S$ denote the set of all small components assigned to clusters in $\mathcal{S}_O \cup \mathcal{S}_N$. Small components currently assigned to clusters in $\mathcal{S}_O \cup \mathcal{S}_N$ are unassigned. Clusters in $\mathcal{S}_O \cup \mathcal{S}_N$ are unmarked and the set of active clusters $\mathcal{S}_A$ is updated by removing clusters in $\mathcal{S}_O$ and adding clusters in $\mathcal{S}_N$.

Next, let $\mathcal{C}'_L$ denote the set of large components which are not assigned to any cluster in $\mathcal{C}_I$. For all clusters $S \in \mathcal{S}_N$, let T_s be the signature assigned to cluster S. For all non-zero T_{sa} where $a \in [c_l]$, a class a large component C is chosen from $\mathcal{C}'_L$, and `Assign-Volume`(S, C) is invoked, concluding the the assignment of large components.

For small components $C \in \mathcal{C}'_S$, `Assign-Small`(C) is invoked. Finally, for all unmarked clusters S in $\mathcal{S}_N \notin \mathcal{S}_M$, `Handle-Unmarked`$(S)$ is invoked. The pseudo code of `Reassign-Large` is deferred to the full version of the paper.

Subroutine `Handle-Deletion`. We present a subroutine `Handle-Deletion` which takes as input a vertex v to be deleted. We assume that Invariant 1 holds prior to the deletion request. Let C denote the component containing v prior to v's deletion s.t. $|C| \in [(1+\frac{\varepsilon}{4})^{i-1}, (1+\frac{\varepsilon}{4})^{i})$ and S_j denote the cluster to which C is assigned. We consider two cases.

Algorithm 6 `Handle-Unmarked`(S_i).

if $A(S_i) = 0$ **then**
  $\mathcal{S}_A \leftarrow \mathcal{S}_A \backslash \{S_i\}, \mathcal{S}_S \leftarrow \mathcal{S}_S \backslash \{S_i\}$.
  if $|\mathcal{S}_S| > 0$ **then** Pick an arbitrary cluster $S \in \mathcal{S}_S$, $\mathcal{S}_M \leftarrow \mathcal{S}_M \backslash \{S\}$.
else
  if ($S_i \in \mathcal{S}_L$ and $\mathcal{S}_S \neq \emptyset$) or ($S_i \in \mathcal{S}_S$ and $|\mathcal{S}_S| > 1$) **then**
    Let $S \neq S_i$ be an unmarked cluster in $\mathcal{S}_S$.
    while $S_i \notin \mathcal{S}_M$ and $\mathcal{S}_S \neq \emptyset$ **do**
      Let C be an arbitrary small component assigned to S.
      if $R_i \geq (1 + \frac{\varepsilon}{4})|C|$ **then**
        `Assign-Volume`(C, S_i).
        `Unassign-Volume`(C, S).
      else
        $\mathcal{S}_M \leftarrow \mathcal{S}_M \cup \{S_i\}$. ▷ S_i is marked
      if $A(S) = 0$ **then**
        $\mathcal{S}_A \leftarrow \mathcal{S}_A \backslash \{S\}$, $\mathcal{S}_S \leftarrow \mathcal{S}_S \backslash \{S\}$. ▷ S is made inactive.
        if $S_i \in \mathcal{S}_L$ and $|\mathcal{S}_S| > 0$ **then**
          Let S be an arbitrary cluster in $\mathcal{S}_S$.
          $\mathcal{S}_M \leftarrow \mathcal{S}_M \backslash \{\mathcal{S}\}$. ▷ S is unmarked.
        if $S_i \in \mathcal{S}_S$ **then**
          if $|\mathcal{S}_S| > 1$ **then**
            Let $S \neq S_i$ be an arbitrary cluster in $\mathcal{S}_S$.
            $\mathcal{S}_M \leftarrow \mathcal{S}_M \backslash \{S\}$. ▷ S is unmarked.
          else
            break. ▷ S_i is the only unmarked cluster in $\mathcal{S}_S$ in this case.

1. C is small and unmarked: Vertex v is removed from C. There are five sub-cases to consider.
   a. If $A(C) = A_{i+1,2}$ and $|C| < A_{i3}$, then $A(C)$ is set to $A_{i+1,1}$.
   b. If $A(C) = A_{i+1,1}$ and $|C| < A_{i2}$, then $A(C)$ is set to A_{i4}.
   c. If $A(C) = A_{i4}$ and $|C| < A_{i1}$, then $A(C)$ is set to A_{i3}.
   d. If $A(C) = A_{i3}$ and $|C| < A_{i0} = (1 + \frac{\varepsilon}{4})^{i-1}$, then C has become a class $i-1$ component. $A(C)$ is set to A_{i2}.
   e. If $C = \emptyset$, $A(C)$ is set to 0.

   In all of the above cases, $A(S_j)$ and R_j are updated. If $S_j \in \mathcal{S}_M$ and $R_j \geq \frac{\varepsilon k}{2}$, S_j is removed from $\mathcal{S}_M$ and `Handle-Unmarked`(S_j) is invoked.
2. C is large or marked: We first remove v from C. We consider the following cases.
   a. C is large and unmarked (i.e. $C \in \mathcal{C}_L \backslash \mathcal{C}_M$): Our algorithm considers the following cases.
      i. If $A(C) = A_{i+1,2}$ and $|C| < A_{i3}$, then $A(C)$ is set to $A_{i+1,1}$.
      ii. If $A(C) = A_{i+1,1}$ and $|C| < A_{i2}$, then $A(C)$ is set to A_{i4}.
      iii. If $A(C) = A_{i4}$ and $|C| < A_{i1}$, then $A(C)$ is set to A_{i3}.
      iv. If $|C| < A_{i0}$, C is marked. If $A(C) = A_{i3}$ then $A(C)$ is set to A_{i2}.

      In the above cases, if $S_j \in \mathcal{S}_M$ and $R_j \geq \frac{\varepsilon k}{2}$, S_j is removed from $\mathcal{S}_M$ and `Handle-Unmarked`(S_j) is invoked.

b. C is marked: There are two cases to consider depending on whether C is large or small (after the deletion of v).

i. Let C be a marked large component of class c where $c = i - c_s$, such that $c \in [c_l - 1]$. If $|C| \geq Dk(1+\frac{\varepsilon}{4})^c - \frac{\varepsilon^2 k}{100}$, there is nothing to be done. Else, if $|C| < Dk(1+\frac{\varepsilon}{4})^c - \frac{\varepsilon^2 k}{100}$, C is unmarked. The counter σ_{c+1} is decremented since C is now considered a class c component, and σ_c is incremented. Then, the subroutine `Reassign-Large` is invoked.

ii. Let C be a marked small component of class c_s. If $|C| \geq Dk - \frac{\varepsilon^2 k}{100}$, nothing is done. Else, if $|C| < Dk - \frac{\varepsilon^2 k}{100}$, C is first unmarked. Then, C is removed from $\mathcal{C}_L$ and added to $\mathcal{C}_S$. The counter σ_1 is decremented. Then, the subroutine `Reassign-Large` is invoked.

The pseudo-code of `Handle-Deletion` is deferred to the full version of the paper.

`Merge-Components`. The subroutine `Merge-Components` is invoked by our algorithm to handle a merge request (u, v). Let C_i, C_j denote the components of u and v respectively. If $C_i = C_j$, nothing needs to be done. Otherwise, w.l.o.g., let $|C_j| \leq |C_i|$ and S_i and S_j denote the clusters on which C_i and C_j are currently assigned. Let $C_m = C_i \cup C_j$.

1. C_m is small: The subroutine `Handle-Small` is called and considers two cases.

a. C_i is marked: We note that C_j cannot be a marked component since this would contradict that C_m is small. This is because if C_j is marked, then $|C_m| \geq 2(Dk - \frac{\varepsilon^2 k}{100}) = Dk + Dk - \frac{\varepsilon^2 k}{50} > Dk$ since $Dk - \frac{\varepsilon^2 k}{50} > \frac{\varepsilon k}{5} - \frac{\varepsilon k}{50} > 0$. Thus, the only possibility is that C_i is marked. Since any marked small component must have volume at least $Dk - \frac{\varepsilon^2 k}{100}$, and C_m is small it follows that $|C_m| \leq Dk$. In this case we note that $A(C_i) = A_{i2} \geq Dk$ so that $A(C_m)$ can be set to $A(C_i)$. C_i is removed from $\mathcal{C}_M$ and C_m is added to $\mathcal{C}_M$. If $|C_m| < Dk$, C_m is added to $\mathcal{C}_M$. As such, C_m is still treated by our algorithm as a large marked component. If $S_i = S_j$ nothing needs to be done. Else, vertices in C_j are migrated to S_i, and `Unassign-Volume`(C_j, S_j) is invoked. Subroutine `Handle-Unmarked`(S_j) is invoked to handle the case when S_j becomes unmarked.

b. C_i and C_j are not marked: In this case, there are two cases to consider.

i. If $A(C_i) \geq |C_m|$, we set $A(C_m)$ to $A(C_i)$ and $A(C_i), A(C_j)$ to 0. If $S_j = S_i$, nothing needs to be done. If $S_j \neq S_i$, vertices in C_j are migrated to S_i, and `Unassign-Volume`(C_j) is invoked. Subroutine `Handle-Unmarked`(S_j) is invoked to handle the case if S_j becomes unmarked.

ii. If $A(C_i) < |C_m|$, we invoke `Unassign-Volume`(C_i, S_i) and `Unassign-Volume`(C_j, S_j). If $R_i \geq (1 + \frac{\varepsilon}{4})|C_m|$, we call `Assign-Volume`$(C_m, S_i)$. If $S_i \neq S_j$, we migrate vertices in C_j to S_i. We invoke `Handle-Unmarked`(S_j) and `Handle-Unmarked`(S_i).
On the other hand, if $R_i < (1 + \frac{\varepsilon}{4})|C|$, S_i is marked and `Assign-Small`(C_m) is invoked.

2. C_m is large: The subroutine `Handle-Large` is invoked which adds C_m to the set of large components.
Let $m' \in [c_l]$ be an integer such that $|C_m| \in [Dk(1 + \frac{\varepsilon}{4})^{m'-1}, Dk(1 + \frac{\varepsilon}{4})^{m'})$. If C_i is marked, i.e. $C_i \in \mathcal{C}_M$, let $i' \coloneqq a + 1$ be the integer where a is an integer such that $|C_i| \in [Dk(1 + \frac{\varepsilon}{4})^a - \frac{\varepsilon^2 k}{100}, Dk(1 + \frac{\varepsilon}{4})^a)$, else we let $i' \coloneqq a$ be the integer where a is an integer such that $|C_i| \in [Dk(1 + \frac{\varepsilon}{4})^{a-1}, Dk(1 + \frac{\varepsilon}{4})^a)$. We consider the following cases.

a. C_j is large or C_j is marked or $m' > i'$: If C_j is marked, we let $j' \coloneqq a + 1$ be the integer where a is an integer such that $|C_j| \in [Dk(1 + \frac{\varepsilon}{4})^a - \frac{\varepsilon^2 k}{100}, Dk(1 + \frac{\varepsilon}{4})^a)$. Else if C_j is unmarked, we let $j' \coloneqq a$ be the integer where a is an integer such that

$|C_j| \in [Dk(1+\frac{\varepsilon}{4})^{a-1}, Dk(1+\frac{\varepsilon}{4})^{a})$. We decrement $\sigma_{j'}$ and $\sigma_{i'}$ by 1, and increment $\sigma_{m'}$ by 1. We remove C_i, C_j from $\mathcal{C}_M$ to make sure that the set of marked components is updated. Thereafter, we update the component sets $\mathcal{C}_S, \mathcal{C}_L$ and call subroutine `Reassign-Large` to handle the assignment of large components.

b. C_j is small and C_j is unmarked and $m' \leq i'$: In this case, we note that $\sigma_{i'}, \sigma_{m'}$ do not change. Thus, the ILP is not solved. We assign volume for the merged component as follows.

In the case when $A(C_i) \geq A(C_m)$, vertices in C_j are migrated to S_i if $S_i \neq S_j$. We invoke `Unassign-Volume`(C_j, S_j), and `Handle-Unmarked`(S_j). Else, nothing needs to be done. The assigned volumes $A(C_i), A(C_j)$ are set to 0. If C_i is marked and $|C_m| < Dk(1+\frac{\varepsilon k}{4})^{i'-1}$, then C_i is removed from $\mathcal{C}_M$ and C_m is added to $\mathcal{C}_M$. We note that in this case, $A(C_i) = A_{i'2} \geq A(C_m)$.

If $A(C_i) < |C_m|$, then we invoke `Unassign-Volume`(C_i, S_i) and `Unassign-Volume`(C_j, S_j).

If $R_i \geq (1+\frac{\varepsilon}{4})|C_m|$, we assign volume for C_m on S_i by invoking `Assign-Volume`(C_m, S_i). If $A(S_j) = 0$, we call `Handle-Unmarked`(S_j). Next, we call `Handle-Unmarked`(S_i). In the case when $A(S_j) \neq 0$ (and $S_j \in \mathcal{S}_A$ as a result), we invoke `Handle-Unmarked`(S_j).

If $R_i < (1+\frac{\varepsilon}{4})|C_m|$, we do the following. While $R_i < (1+\frac{\varepsilon}{4})|C_m|$, we call `Unassign-Volume`$(C, S_i)$ where C is an arbitrary small component assigned on S_i, and add it to a set $\mathcal{C}'_S$. Once $R_i \geq (1+\frac{\varepsilon}{4})|C_m|$, we invoke `Assign-Volume`$(C_m, S_i)$ and migrate vertices in C_j to S_i if needed (in the case when $S_i \neq S_j$) together with a call to `Unassign-Volume`(C_j, S_j). Next, we assign components in $\mathcal{C}'_S$ to S_i as long as there is sufficient volume and S_i is not marked. If all components in $\mathcal{C}'_S$ have been assigned and S_i is unmarked, we call `Handle-Unmarked`(S_i).

If S_j is unmarked, we invoke `Handle-Unmarked`(S_j). All remaining unassigned components C in $\mathcal{C}'_S$ (if any) are assigned by invoking `Assign-Small`(C). Finally, the component sets $\mathcal{C}_S, \mathcal{C}_L$ are updated.

The pseudo-code of subroutines `Merge-Components`, `Handle-Small` and `Handle-Large` are deferred to the full version of the paper.

We present the pseudo-code of our final online algorithm `OBA` that utilizes the aforementioned subroutines as follows.

Algorithm 7 `OBA`.

$\mathcal{C}_S, \mathcal{C}_L, \mathcal{C}_M, \mathcal{C}_C \leftarrow \emptyset$.
for all $S_i \in \mathcal{S}$ **do** $A(S_i) \leftarrow 0, R_i = (1+\varepsilon)k$
$\mathcal{S}_S, \mathcal{S}_L, \mathcal{S}_M, \mathcal{S}_A \leftarrow \emptyset$.
for r_t at time t **do**
    if r_t is an insertion request for vertex v **then**
        `Handle-Insertion`(v).
    else if r_t is a deletion request for vertex v **then**
        `Handle-Deletion`(v).
    else
        Let r_t be a merge request between vertices (u, v).
        `Merge-Components`(u, v).

3 Analysis of OBA

In this section, we give a proof sketch towards $O(\log k)$-competitiveness of our algorithm. Technical details of our proof, together with proofs of correctness and resource competitiveness are deferred to the full version of the paper.

Our analysis is intricate and relies on various charging arguments. At a high level, we show that a vertex can be assigned sufficient credit on insertion such that the total amount of credit on it at any point in time is sufficient to pay for the cost charged against it. The total credit on a vertex is the sum of its *direct* and *indirect* credits. *Direct* credit is a one-time credit assigned to v on its insertion. *Indirect* credit on v is credit passed on to it by other vertices which get deleted from v's component.

Warm up: An insertions-only case. Let us first consider a (finite) request sequence σ which only consists of insertion and merge requests. For this sequence, the assigned volume of any component containing a vertex v monotonically increases over time, and no components are ever marked. We show that it suffices to charge each vertex an amount $O(\log k + \frac{f(\varepsilon)}{\varepsilon^3})$ on its insertion. On v's insertion, our algorithm assigns $\{v\}$ to an active cluster in $\mathcal{S}_A$, or makes a new cluster active and assigns $\{v\}$ to it. On a merge request between components C_i and C_j which are merged into a *small* component C_m, where w.l.o.g., $|C_i| \geq |C_j|$ our algorithm always migrates vertices in C_j to C_i (in the case when $S_i \neq S_j$) as long as the cluster S_i containing C_i has sufficient residual volume. The first key observation is that for such a merge, all vertices in C_j are now part of a component (i.e. C_m) with a larger component class. Thus, if $A(C_i) \geq |C_m|$ or $R_i \geq (1 + \frac{\varepsilon}{4})|C_m|$, only vertices in C_j are migrated. Potentially some other vertices may be migrated to S_j if S_j becomes unmarked after volume is un-assigned due to the migration of C_j (we ignore these costs for now). For any vertex v, and the component C containing v, note that C's component class increases monotonically over time due to merge requests. The number of component classes, which is $O(c_s + c_l) = O(\log k + \frac{1}{\varepsilon^2})$ bounds the number of times the component class can increase.

On the other hand, if R_i is insufficient to accommodate vertices in C_j (i.e. in the case when $A(C_i) < |C_m|$ and $R_i < (1 + \frac{\varepsilon}{4})k$), vertices in both C_i and C_j are migrated to another cluster. Note that in this case, $A(C_m) > A(C_i)$. Since there are at most $4(c_s + c_l) = O(\log k + \frac{1}{\varepsilon^2}) = O(\log k)$ possible values that assigned volumes can take, and assigned volume increases monotonically for any component, each vertex v can migrate only $O(\log k + \frac{1}{\varepsilon^2})$ times in this manner.

Crucially, we note that if the component C containing v becomes large (or C is large and its component class increases), a total of $O(f(\varepsilon)k)$ migration cost may be incurred, by Lemma 10. This migration cost can be incurred a total of $c_l = O(\frac{1}{\varepsilon^2})$ times. By definition, a C has size at least $Dk \geq \frac{\varepsilon k}{5}$; we charge the migration cost to all $\Omega(\varepsilon k)$ vertices in C. Thus, each vertex in C contributes at least $O(\frac{f(\varepsilon)}{\varepsilon})$ credit a total of $O(\frac{1}{\varepsilon^2})$ times. Therefore, a credit of $O(\frac{f(\varepsilon)}{\varepsilon^3})$ on a vertex is sufficient to pay for migrations of this type.

We now analyze the potential migration costs incurred when a component C migrates from one cluster (say S_i) to S_j, which may lead to a call to `Handle-Unmarked`(S_i) leading to components migrations to S_i. To account for these costs, a cluster credit $\Theta(|C|)$ is left on S_i when C migrates to S_j, and each vertex in C is charged an amount $O(1)$. As a result, each time the residual volume of a cluster increases, there is enough cluster credit to pay for successive migrations (as part of `Handle-Unmarked`). Note that in subroutine `Handle-Unmarked`, only small components are migrated from an unmarked cluster, and the total amount of migrated components is proportional to the increase in residual volume.

Thus, a vertex is charged for cluster credit each time its component is promoted to a higher class or its assigned volume increases. Since the number of choices for assigned volumes (and component classes) is $O(\log k + \frac{1}{\varepsilon^2})$, any vertex is charged $O(\log k + \frac{1}{\varepsilon^2})$ in this manner.

By the above, it follows that assigning a one-time direct credit of $O(\frac{1}{\varepsilon^3} f(\varepsilon) + \log k)$ on each vertex v on its insertion is sufficient. Since OPT is lower bounded by the number of vertex insertions, this yields $O(\log k)$ competitiveness.

The General Case. Let us consider the case when the request sequence consists of insertions, merges and deletions. In this case, the component class of any component C and its assigned volume $A(C)$ do not necessarily increase over time, in general. Thus, direct credit on v is insufficient, and our analysis crucially relies on *indirect* credit. Indirect credit is assigned to any vertex in the following manner: whenever a vertex u is deleted from v's component C, an amount of indirect credit is distributed to all remaining vertices in C. Some credit is also left as cluster credit on the cluster containing C on u's deletion. Cluster credit is used to pay for re-balancing and migrating components as in the previous case. The total indirect credit assigned to vertices in C, together with cluster credit that is left on u's deletion is charged against vertex u. On the other hand, the total credit assigned to a vertex is the sum of the direct credit and indirect credit distributed to it by other vertices. Let C denote the component containing v. We show that even in the case of deletions, it suffices to assign a direct credit of $O(\frac{1}{\varepsilon^3} f(\varepsilon) + \log k)$ to vertex v upon insertion, which is used to pay for:

1. the unit migration cost incurred each time C is promoted to a component class for the *first* time, or the assigned volume of C increases for the first time as in the previous case.
2. indirect credit assigned to vertices in C upon the deletion of v.
3. cluster credit assigned to the cluster containing C, upon v's deletion.
4. cluster credit assigned to the cluster containing C, before C is migrated to another cluster as a result of C being promoted to either a component class for the first time, or an increase in its assigned volume for the first time.

Consider a vertex v and let C denote an (unmarked) component containing v with assigned volume $A(C)$. Upon vertex deletions from C, $A(C)$ may decrease. Suppose $A(C) = A_{ij}$ for some $j \in \{1, 2, 3, 4\}$. Consider a set of deleted vertices C_D which lead to a reduction in C's assigned volume to $A_{i,j-1}$. For this to happen, $|C_D| \geq \frac{\varepsilon}{16}(1 + \frac{\varepsilon}{4})^{i-1}$. On the other hand, $|C| \leq (1 + \frac{j-1}{16}\varepsilon)(1 + \frac{\varepsilon}{4})^{i-1} \leq (1 + \frac{3\varepsilon}{16})(1 + \frac{\varepsilon}{4})^{i-1}$, after vertices in C_D are deleted. For each vertex u in C_D, we distribute a total indirect credit of $O(\frac{1}{\varepsilon^2} f(\varepsilon))$ uniformly to all remaining vertices in C. As a result, the total indirect credit gained by each remaining vertex in $C \backslash C_D$ is at least $O(|C_D| \cdot \frac{O(\frac{f(\varepsilon)}{\varepsilon^2})}{(1+\frac{3\varepsilon}{16})(1+\frac{\varepsilon}{4})^{i-1}}) = O(\frac{f(\varepsilon)}{\varepsilon})$ after all vertices in C_D are deleted. The *indirect* credit gained by any vertex v in $C \backslash C_D$ is used to pay for: i) a single *future* migration of a vertex $v \in C \backslash C_D$ which leads to an increase in either the assigned volume of v's component or the component class of v's component and ii) leave $O(1)$ cluster credit when the aforementioned migration happens.

Next, consider the time when a component C is marked, such that $|C| < Dk(1 + \frac{\varepsilon}{4})^{i-1}$ where $i \geq 1$. Once $|C| < Dk(1 + \frac{\varepsilon}{4})^{i-1} - \frac{\varepsilon^2 k}{100}$, this leads to either a call to `Reassign-Large` (if C is large) or C is treated as a small component thereafter (see `Handle-Deletion`). Since C has size at most k, this implies that at least $\frac{\varepsilon^2}{100} = \Omega(\varepsilon^2)$ fraction of vertices in C are deleted before either of these events happen. The total migration and reassignment cost incurred due to `Reassign-Large` is $O(f(\varepsilon)k)$ by Lemma 10. To pay for this cost, leave an indirect credit of $O(\frac{f(\varepsilon)}{\varepsilon})$ on each remaining vertex in C, and assign a cluster credit of $O(1)$, it suffices to charge each deleted vertex from C an amount $O(\frac{1}{\varepsilon^3} f(\varepsilon))$. The cluster credit is used to pay for the migration cost of vertices to S_i if S_i becomes unmarked. The amount charged against a deleted vertex u is paid by the direct credit assigned to u upon its insertion.

By the above, it follows that assigning a direct credit of $O(\log k + \frac{f(\varepsilon)}{\varepsilon^3})$ to a vertex upon its insertion is sufficient. Since OPT is lower bounded by the number of vertices inserted over all time, this yields $O(\log k)$ competitiveness for constant $\varepsilon > 0$.

4 OBADC with Predictions

In this section, we present our algorithm for OBADC which is augmented with machine learned predictions. The problem formulation and objectives are identical as in the standard un-augmented setup. At each time step t, a request r_t which corresponds to either a vertex insertion, deletion or a merge between two components is issued. At any given point in time, an online algorithm is required to maintain an assignment of components to clusters, while respecting the over-provisioned cluster capacity of $(1+\varepsilon)k$, such that all vertices in any component are assigned to the same cluster. In light of the $\Omega(\log k)$ lower bound on the competitiveness of any randomized algorithm, it is natural to ask whether this barrier can be circumvented given oracle access to machine-learned predictions to guide the assignment of components. To this end, we recap our prediction model.

Prediction Model. Let V_{t-1} denote the set of vertices at the beginning of time $t-1$. For a request r_t at time t which corresponds to an insertion of vertex v, we obtain:

1. with probability $\eta > 0$, a predicted set $P(v)$ of all vertices in V_{t-1} which will be in the same component as v.
2. with probability $1-\eta$, a *null* prediction, i.e. $P(v) = \emptyset$. Thus no information is revealed on the insertion of v.

We present an algorithm `Predicted-OBA` which is $O(1)$-consistent and $\min\{\log \frac{1}{\eta}, \log k\}$-robust. This algorithm is similar in many ways to our $O(\log k)$ competitive algorithm `OBA`. We utilize identical data structures, definitions and subroutines as detailed in Section 2. We also maintain Invariants 1 through 4 as in Section 2.

Our algorithm `Predicted-OBA` is as follows.

Algorithm 8 `Predicted-OBA`.

$\mathcal{C}_S, \mathcal{C}_L, \mathcal{C}_M, \mathcal{C}_C \leftarrow \emptyset$.
for all $S_i \in \mathcal{S}$ **do** $A(S_i) \leftarrow 0, R_i = (1+\varepsilon)k$
$\mathcal{S}_S, \mathcal{S}_L, \mathcal{S}_M, \mathcal{S}_A \leftarrow \emptyset$.
for r_t at time t **do**
    if r_t is an insertion request for vertex v **then**
        `Predicted-Insertion`$(v, P(v))$.
    else if r_t is a deletion request for vertex v **then**
        `Handle-Deletion`(v).
    else
        Let r_t be a merge request between components C_i and C_j.
        `Merge-Components`(C_i, C_j).

We give an additional subroutine, `Predicted-Insertion` as follows.

Subroutine `Predicted-Insertion`. This subroutine takes as input a vertex v inserted at time t and a predicted set $P(v)$ consisting of vertices at time t that will be in the same component as v. If $P(v) = \emptyset$, the subroutine `Handle-Insertion`(v) is invoked. Else, a set σ_I of merge requests is created where $\sigma_I = \{(u, v) |\ u \in P(v)\}$. For each merge request in σ_I, the subroutine `Merge-Components`(u, v) is invoked.

The following lemma bounds the competitive ratio of Algorithm `Predicted-OBA`.

▶ **Lemma 11.** *Algorithm `Predicted-OBA` is $O(\min\{\log(1/\eta), \log k\})$-competitive.*

We build towards a proof of Lemma 11 in the following. We first analyze the probability of a merge request being incident to vertices in distinct components C_i, C_j as a function of component sizes.

▶ **Lemma 12.** *Let r_t be a merge request between components C and C' at time t, such that $s = |C| \leq |C'|$ without loss of generality. Then, the probability that $C' \neq C$ is bounded by $e^{-\eta s}$.*

Proof. We note that if $C \neq C'$, then for at least $s = |C|$ insertions of vertices w, $P(w) = \emptyset$. Since the probability that $P(w) = \emptyset$ for any w is $(1-\eta)$, it follows that for s vertex insertions, the probability that all predicted sets are empty is bounded by $(1-\eta)^s \leq e^{-\eta s}$. ◀

We define monotonic merges as follows.

▶ **Definition 13** (Monotonic Merge)**.** *Let C be the component containing vertex v at any point in time throughout v's lifetime. Let t_1 and t_2 be time-steps s.t. v is inserted at time t_1 and deleted at time t_2. A merge request involving components C and C' at time $t \in [t_1, t_2)$ is said to be a monotonic merge for v if $|C| \leq |C'|$ and one of the following holds:*

1. *The assigned volume of C is set to A_{ij} by our algorithm where $i \in [c_s + c_l], j \in \{1, 2, 3, 4\}$ for the first time during time interval $[t_1, t_2)$.*
2. *The component class of C is set to $i \in [c_s + c_l]$ by our algorithm for the first time during time interval $[t_1, t_2)$.*

A merge request for v is non-monotonic if it is not monotonic.

▶ **Lemma 14.** *The expected number of monotonic merge requests for any vertex v including merge requests created by Subroutine `Predicted-Insertion` is bounded by $O(\log(\frac{1}{\eta}))$.*

Proof. Let C denote the component of v, such that v is inserted at time t_1 and deleted at time t_2. Let $d = (1 + \frac{\varepsilon}{4})$, so that a class i component has volume in $[d^{i-1}, d^i)$. By Lemma 12, the probability that a merge request involving C where C is the smaller of the two components and belongs to class i, is bounded by $e^{-\eta \cdot d^{i-1}}$. Thus, the expected number of monotonic merge requests for v is bounded by

$$\sum_{i=0}^{\log k} e^{-d^i \eta} = \sum_{i=0}^{\log(1/\eta)} e^{-d^i \eta} + \sum_{i=\log(1/\eta)+1}^{\log k} e^{-d^i \eta} \leq \log(1/\eta) + O(1) = O(\log(1/\eta)).$$ ◀

Proof Sketch of Lemma 11. Let us recall the analysis of $O(\log k)$-competitiveness of our algorithm `OBA`. We argued that it suffices to leave a direct credit of $O(\log k + \frac{f(\varepsilon)}{\varepsilon^3})$ on any vertex v upon its insertion, which is sufficient to pay for: i) migration costs and cluster credits charged against v for both monotonic and non-monotonic merges and, ii) cluster and indirect credits charged against v upon its deletion. The $O(\log k)$ term comes from the fact a vertex v can be part of monotonic merges for a total of $c_s + c_l = O(\log k)$ times, and is charged $O(\log k)$ times for merges while C is small. On the other hand, for a non-monotonic merge, v is charged an amount $O(1)$ to pay for its migration cost and leave cluster credit if the residual volume of the cluster C is assigned to, is no longer sufficient to accommodate an increase in $A(C)$. Since the number of choices for assigned volumes is $O(c_s + c_l) = O(\log k + \frac{1}{\varepsilon^2})$, the total cost charged against v for non-monotonic merges is $O(\log k + \frac{1}{\varepsilon^2})$.

We modify our charging argument for non-monotonic merges as follows. Ignoring the reassignment of components that takes place as a result of subroutine `Handle-Unmarked` for the sake of clarity, we crucially observe that whenever the residual volume of C's cluster (denoted by S_i) decreases, it is due to either: i) vertices in a smaller component involved in a merge request migrate to S_i, or ii) new vertices are inserted to S_i's cluster. We modify our charging argument such that the migrated or inserted vertex in the above cases leaves $O(1)$ extra cluster credit on S_i. Thus, whenever the residual volume of S_i decreases by $\Omega(k)$, these extra cluster credits can be used to pay for non-monotonic merges involving vertices already assigned on S_i. As a result, we focus on the cost charged for monotonic merges in the remainder of the analysis.

By Lemma 14, it follows that the expected number of monotonic merges for a vertex is bounded by $O(\log(\frac{1}{\eta}))$. By our analysis of `OBA`, it follows that the expected cost charged to a vertex for monotonic merges is $O(\frac{f(\varepsilon)}{\varepsilon})$. Thus, an expected credit of $O(\log(1/\eta)\frac{f(\varepsilon)}{\varepsilon})$ is sufficient for monotonic merges. On the other hand, when a vertex v is deleted, we charge an amount $O(\frac{f(\varepsilon)}{\varepsilon^3})$ to it which is assigned as indirect credit and cluster credit. As a result, it follows that a total direct credit of $O(\log(1/\eta)\frac{f(\varepsilon)}{\varepsilon} + \frac{f(\varepsilon)}{\varepsilon^3})$ in expectation, is sufficient to pay for all costs charged against v throughout its lifetime.

This yields $O(\log(1/\eta))$-competitiveness. On the other hand, as $\eta \to 0$, the behavior of algorithm `Predicted-OBA` resembles the behavior of algorithm `OBA`, and thus, the competitiveness of `Predicted-OBA` is bounded by $O(\log k)$ in the worst-case. This yields $O(\min(\log(1/\eta), \log k))$-competitiveness. The following lemma establishes $O(1)$ competitiveness when $\eta = 1$. The proof is deferred to the full version of the paper.

▶ **Lemma 15.** *Algorithm `Predicted-OBA` is $O(1)$ consistent.*

Theorem 2 follows from Lemmas 11 and 15.

5 Future Work

Our model for OBADC is restrictive in the sense that component sizes decrease only as a result of vertex deletions; in particular, our model does not allow for splitting of existing components. We believe that such a model will be much more challenging to handle; indeed, it could be considered as a generalization of the general model of OBGR, for which there is a significant gap between the upper and lower bounds for the best competitive ratio achievable.

In this work, we introduced the OBADC and studied it in the context of machine-learned predictions. It would be interesting to extend our prediction model to the case when predictions are always available but may be erroneous. Other prediction models for online balanced allocation problems would be worth exploring.

References

1 Khaled Almi'ani, Young Choon Lee, and Bernard Mans. Resource demand aware scheduling for workflows in clouds. In *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, pages 1–5. IEEE, 2017.

2 Chen Avin, Marcin Bienkowski, Andreas Loukas, Maciej Pacut, and Stefan Schmid. Dynamic balanced graph partitioning. *SIAM Journal on Discrete Mathematics*, 34(3):1791–1812, 2020. doi:10.1137/17M1158513.

3 Yossi Azar, Chay Machluf, Boaz Patt-Shamir, and Noam Touitou. Competitive Vertex Recoloring. In Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 13:1–13:20, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2022.13.

4 Marcin Bienkowski, Martin Böhm, Martin Koutecký, Thomas Rothvoß, Jiří Sgall, and Pavel Veselý. Improved analysis of online balanced clustering. In *Approximation and Online Algorithms: 19th International Workshop, WAOA 2021, Lisbon, Portugal, September 6–10, 2021, Revised Selected Papers*, pages 224–233, Berlin, Heidelberg, 2021. Springer-Verlag. doi:10.1007/978-3-030-92702-8_14.

5 Jan van den Brand, Sebastian Forster, Yasamin Nazari, and Adam Polak. On dynamic graph algorithms with predictions. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3534–3557. SIAM, 2024. doi:10.1137/1.9781611977912.126.

6 Prasad Calyam, Sripriya Seetharam, Baisravan Homchaudhuri, and Manish Kumar. Resource defragmentation using market-driven allocation in virtual desktop clouds. In *2015 IEEE International Conference on Cloud Engineering*, pages 246–255. IEEE, 2015. doi:10.1109/IC2E.2015.37.

7 Jiuxin Cao, Zhuo Ma, Jue Xie, Xiangying Zhu, Fang Dong, and Bo Liu. Towards tenant demand-aware bandwidth allocation strategy in cloud datacenter. *Future Generation Computer Systems*, 105:904–915, 2020. doi:10.1016/J.FUTURE.2017.06.005.

8 Vincent Cohen-Addad, Tommaso d'Orsi, Anupam Gupta, Euiwoong Lee, and Debmalya Panigrahi. Max-cut with *epsilon*-accurate predictions. *arXiv preprint arXiv:2402.18263*, 2024.

9 Hancong Duan, Chao Chen, Geyong Min, and Yu Wu. Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems. *Future Generation Computer Systems*, 74:142–150, 2017. doi:10.1016/J.FUTURE.2016.02.016.

10 Tobias Forner, Harald Räcke, and Stefan Schmid. Online balanced repartitioning of dynamic communication patterns in polynomial time. In *Symposium on Algorithmic Principles of Computer Systems (APOCS)*, pages 40–54, 2021. doi:10.1137/1.9781611976489.4.

11 Anupam Gupta, Debmalya Panigrahi, Bernardo Subercaseaux, and Kevin Sun. Augmenting online algorithms with ε-accurate predictions. *Advances in neural information processing systems*, 35:2115–2127, 2022.

12 Sijin He, Li Guo, and Yike Guo. Real time elastic cloud management for limited resources. In *2011 IEEE 4th International Conference on Cloud Computing*, pages 622–629. IEEE, 2011. doi:10.1109/CLOUD.2011.47.

13 Monika Henzinger, Stefan Neumann, Harald Räcke, and Stefan Schmid. *Tight Bounds for Online Graph Partitioning*, pages 2799–2818. Society for Industrial and Applied Mathematics, USA, 2021.

14 Monika Henzinger, Stefan Neumann, and Stefan Schmid. Efficient distributed workload (re-)embedding. In *Abstracts of the 2019 SIGMETRICS/Performance Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '19, pages 43–44, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3309697.3331503.

15 Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. *Journal of the ACM (JACM)*, 68(4):1–25, 2021. doi:10.1145/3447579.

16 Zoltán Ádám Mann. Allocation of virtual machines in cloud data centers—a survey of problem models and optimization algorithms. *Acm Computing Surveys (CSUR)*, 48(1):1–34, 2015. doi:10.1145/2797211.

17 Suhib Bani Melhem, Anjali Agarwal, Nishith Goel, and Marzia Zaman. Markov prediction model for host load detection and vm placement in live migration. *IEEE Access*, 6:7190–7205, 2017. doi:10.1109/ACCESS.2017.2785280.

18 Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. *Communications of the ACM*, 65(7):33–35, 2022. doi:10.1145/3528087.

19 Saloua El Motaki, Ali Yahyaouy, and Hamid Gualous. A prediction-based model for virtual machine live migration monitoring in a cloud datacenter. *Computing*, 103(11):2711–2735, 2021. doi:10.1007/S00607-021-00981-3.

20 Maciej Pacut, Mahmoud Parham, and Stefan Schmid. Optimal online balanced partitioning. In *INFOCOM 2021*, 2021.

21 Cynthia A. Phillips, Cliff Stein, Eric Torng, and Joel Wein. Optimal time-critical scheduling via resource augmentation (extended abstract). In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '97, pages 140–149, New York, NY, USA, 1997. Association for Computing Machinery. doi:10.1145/258533.258570.

22 Rajmohan Rajaraman and Omer Wasim. Improved bounds for online balanced graph repartitioning. In *30th Annual European Symposium on Algorithms (ESA 2022)*, 2022. doi:10.4230/LIPIcs.ESA.2022.83.

23 Rajmohan Rajaraman and Omer Wasim. Competitive capacitated online recoloring. In *32nd Annual European Symposium on Algorithms (ESA 2024)*, 2024. doi:10.4230/LIPIcs.ESA.2024.95.

24 Dhruv Rohatgi. Near-optimal bounds for online caching with machine learned advice. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1834–1845. SIAM, 2020. doi:10.1137/1.9781611975994.112.

25 Tim Roughgarden. *Beyond the worst-case analysis of algorithms*. Cambridge University Press, 2021.

26 A. Schrijver. Theory of linear and integer programming. In *Wiley-Interscience series in discrete mathematics and optimization*, 1999.

27 Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

28 Jaspreet Singh and Navpreet Kaur Walia. A comprehensive review of cloud computing virtual machine consolidation. *IEEE Access*, 11:106190–106209, 2023. doi:10.1109/ACCESS.2023.3314613.

29 Daniel D. Sleator and Robert E. Tarjan. Amortized efficiency of list update and paging rules. *Commun. ACM*, 28(2):202–208, February 1985. doi:10.1145/2786.2793.

30 Mukundan Sridharan, Prasad Calyam, Aishwarya Venkataraman, and Alex Berryman. Defragmentation of resources in virtual desktop clouds for cost-aware utility-optimal allocation. In *2011 Fourth IEEE International Conference on Utility and Cloud Computing*, pages 253–260. IEEE, 2011. doi:10.1109/UCC.2011.41.

31 Ruitao Xie, Xiaohua Jia, Kan Yang, and Bo Zhang. Energy saving virtual machine allocation in cloud computing. In *2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops*, pages 132–137. IEEE, 2013. doi:10.1109/ICDCSW.2013.37.

32 Hiroki Yanagisawa, Takayuki Osogami, and Rudy Raymond. Dependable virtual machine allocation. In *2013 Proceedings IEEE INFOCOM*, pages 629–637. IEEE, 2013. doi:10.1109/INFCOM.2013.6566848.

33 N. Young. The k-server dual and loose competitiveness for paging. *Algorithmica*, 11(6):525–541, June 1994. doi:10.1007/BF01189992.

34 Neal E. Young. On-line file caching. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '98, pages 82–86, USA, 1998. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=314613.314658.

Tight Bounds on List-Decodable and List-Recoverable Zero-Rate Codes

Nicolas Resch
Informatics' Institute, University of Amsterdam, The Netherlands

Chen Yuan
School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China

Yihan Zhang
Institute of Science and Technology Austria, Klosterneuburg, Austria

Abstract

In this work, we consider the list-decodability and list-recoverability of codes in the *zero-rate regime*. Briefly, a code $\mathcal{C} \subseteq [q]^n$ is (p, ℓ, L)-list-recoverable if for all tuples of input lists $(Y_1, \ldots, Y_n)$ with each $Y_i \subseteq [q]$ and $|Y_i| = \ell$, the number of codewords $c \in \mathcal{C}$ such that $c_i \notin Y_i$ for at most pn choices of $i \in [n]$ is less than L; list-decoding is the special case of $\ell = 1$. In recent work by Resch, Yuan and Zhang (ICALP 2023) the zero-rate threshold for list-recovery was determined for all parameters: that is, the work explicitly computes $p_* := p_*(q, \ell, L)$ with the property that for all $\varepsilon > 0$ (a) there exist positive-rate $(p_* - \varepsilon, \ell, L)$-list-recoverable codes, and (b) any $(p_* + \varepsilon, \ell, L)$-list-recoverable code has rate 0. In fact, in the latter case the code has *constant* size, independent on n. However, the constant size in their work is quite large in $1/\varepsilon$, at least $|\mathcal{C}| \geqslant (\frac{1}{\varepsilon})^{O(q^L)}$.

Our contribution in this work is to show that for all choices of q, ℓ and L with $q \geqslant 3$, any $(p_* + \varepsilon, \ell, L)$-list-recoverable code must have size $O_{q,\ell,L}(1/\varepsilon)$, and furthermore this upper bound is complemented by a matching lower bound $\Omega_{q,\ell,L}(1/\varepsilon)$. This greatly generalizes work by Alon, Bukh and Polyanskiy (IEEE Trans. Inf. Theory 2018) which focused only on the case of binary alphabet (and thus necessarily only list-decoding). We remark that we can in fact recover the same result for $q = 2$ and even L, as obtained by Alon, Bukh and Polyanskiy: we thus strictly generalize their work.

Our main technical contribution is to (a) properly define a linear programming relaxation of the list-recovery condition over large alphabets; and (b) to demonstrate that a certain function defined on a q-ary probability simplex is maximized by the uniform distribution. This represents the core challenge in generalizing to larger q (as a binary simplex can be naturally identified with a one-dimensional interval). We can subsequently re-utilize certain Schur convexity and convexity properties established for a related function by Resch, Yuan and Zhang along with ideas of Alon, Bukh and Polyanskiy.

2012 ACM Subject Classification Mathematics of computing → Coding theory

Keywords and phrases List Decoding, List Recovery, Zero Rate

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.82

Related Version *Full Version*: https://arxiv.org/abs/2309.01800

Funding The research of C. Yuan was support in part by the National Key R&D Program of China under Grant 2023YFE0123900 and Natural Science Foundation of Shanghai under the 2024 Shanghai Action Plan for Science, Technology and Innovation Grant 24BC3200700. The research of N. Resch is supported in part by an NWO (Dutch Research Council) grant with number C.2324.0590, and this work was done in part while he was visiting the Simons Institute for the Theory of Computing, supported by DOE grant #DE-SC0024124.

1 Introduction

Given an error-correcting code $\mathcal{C} \subseteq [q]^n$, a fundamental requirement is that the codewords are sufficiently well-spread in order to guarantee some non-trivial correctability properties. This is typically enforced by requiring that the minimum distance of the code $d = \min\{d_{\mathrm{H}}(\boldsymbol{c}, \boldsymbol{c}') :$


© Nicolas Resch, Chen Yuan, and Yihan Zhang;
licensed under Creative Commons License CC-BY 4.0



$\boldsymbol{c} \neq \boldsymbol{c}' \in \mathcal{C}\}$, where $d_{\mathrm{H}}(\cdot,\cdot)$ denotes the Hamming distance (i.e. the number of coordinates on which two strings differ). Note that minimum distance d is equivalent to the following "packing" property: if we put a ball of radius $r := \lfloor d/2 \rfloor$ around any point $\boldsymbol{z} \in [q]^n$ – i.e. we consider the Hamming ball $\mathcal{B}_{\mathrm{H}}(\boldsymbol{y}, r) := \{\boldsymbol{x} \in [q]^n : d_{\mathrm{H}}(\boldsymbol{x}, \boldsymbol{y}) \leqslant r\}$ – then all these balls contain at most 1 codeword from $\mathcal{C}$.

This latter viewpoint can easily be generalized to obtain *list-decodability*, where we now require that such Hamming balls do not capture "too many" codewords. That is, for $p \in [0,1]$ and $L \in \mathbb{N}$ a code is called (p, L)-*list-decodable* if every Hamming ball of radius pn contains less than L codewords from $\mathcal{C}$. In notation: for all $\boldsymbol{y} \in [q]^n$, $|\mathcal{B}_{\mathrm{H}}(\boldsymbol{y}, pn) \cap \mathcal{C}| \leqslant L - 1$.[1] This notion was already introduced in the 50's by Elias and Wozencraft [10, 30, 11] but has in the past 20 years seen quite a bit of attention due to its connections to other parts of theoretical computer science [13, 2, 21, 19, 18, 26].

One can push this generalization further to obtain *list-recoverability*. Here, we consider a tuple of input lists $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$, where each $Y_i \subseteq [q]$ has size at most ℓ (for some $\ell \in \mathbb{N}$). The requirement is that the number of codewords that "disagree" with $\boldsymbol{Y}$ in at most pn coordinates is at most $L - 1$. More formally, if for all $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ the number of codewords $\boldsymbol{c} \in \mathcal{C}$ such that $|\{i \in [n] : c_i \notin Y_i\}| \leqslant pn$ is at most $L - 1$, the code is called (p, ℓ, L)-*list-recoverable*. Note that (p, L)-list-decodability is nothing other than $(p, 1, L)$-list-recoverability. Initially, list-recoverability was abstracted as a useful stepping stone towards list-decoding concatenated codes. However, in recent years this notion has found many connections to other parts of computer science, e.g. in cryptography [15, 16], randomness extraction [14], hardness amplification [8], group testing [17, 23], streaming algorithms [9], and beyond.

Rate versus noise-resilience. Having fixed a desired "error-tolerance" as determined by the parameters p, ℓ and L we would also like the code $\mathcal{C}$ to be as large as possible: intuitively, this implies that the code contains the minimal amount of redundancy possible. A fundamental question in coding theory is to understand the achievable tradeoffs between the rate $R := \frac{\log_q |\mathcal{C}|}{n}$ and some "error-resilience" property of the code, e.g., minimum distance, list-decodability, or list-recoverability.

This question in full generality is wide open. Even for the special case of $q = 2$ and $L = 2$ (i.e. determining the optimal tradeoff between rate and distance for binary codes) is unclear: on the possibility side we have the Gilbert-Varshamov bound [12, 28] showing $R \geqslant 1 - H_2(p/2)$ is achievable (here, $H_2(x) = -x \log_2 x - (1-x)\log_2(1-x)$ is the binary entropy function), while bounds of Elias and Bassalygo [3] and the linear programming bound [29, 22, 7] give incomparable and non-tight upper bounds. None of these bounds have been substantially improved in at least 40 years. The situation is even more complicated for larger q: for $q = 49$ (and larger prime powers) the celebrated algebraic geometry codes of Tsafsman, Vladut and Zink [27] provide explicit codes of higher rate in certain regimes than those promised by the Gilbert-Varshamov bound.

When one relaxes the question to allow an asymptotically growing list size L then we do have a satisfactory answer: the answer is provided by the list-decoding/-recovery theorem, which states that for all $\varepsilon > 0$ there exist $(p, \ell, O(1/\varepsilon))$-list-recoverable codes of rate $1 - H_{q,\ell}(p)$ where

[1] Typically the upper bound is L, rather than $L - 1$. However, for "impossibility" arguments this parametrization is more common, as it leads to less cumbersome computations.

$$H_{q,\ell}(x) := p \log_q \left(\frac{q-\ell}{p} \right) + (1-p) \log_q \left(\frac{\ell}{1-p} \right)$$

is (q, ℓ)-ary entropy function [24].[2] On the other hand, any code of rate $R \geqslant 1 - h_{q,\ell}(p)$ fails to be (p, ℓ, L)-list-recoverable unless $L \geqslant q^{\Omega(\varepsilon n)}$. However, this does not provide very meaningful bounds if one is interested in, say, $(p, 2, 5)$-list-recoverable codes.

Positive versus zero-rate regimes. Thus far, we have implicitly been discussing the *positive-rate regime*. However, one can also ask questions about the behaviour of codes in the *zero-rate regime*. For context, recent work by Resch, Yuan and Zhang [25] computed the zero-rate threshold for list-recovery: that is, for all alphabet sizes $q \geqslant 2$, input list sizes ℓ and output list size L, they determine the value $p_*(q, \ell, L)$ such that (a) for all $p < p_*(q, \ell, L)$ there exist infinite families of positive rate (p, ℓ, L)-list-recoverable codes over the alphabet $[q]$, and (b) for all $p > p_*(q, \ell, L)$ there does not exist such an infinite family.

Having now delineated the "positive rate" and the "zero-rate" regimes depending on how p compares to $p_*(q, \ell, L)$, in this work we study the zero-rate regime for list-recoverable codes for all alphabet sizes q. In [25], it is shown that (p, ℓ, L)-list-recoverable codes $\mathcal{C} \subseteq [q]^n$ with $p = p_*(q, \ell, L) + \varepsilon$ have constant size (that is, independent of the block length n); however, this constant is massive in the parameters due to the use of a Ramsey-theoretic bound. In particular, the dependence on ε is at least $(1/\varepsilon)^{2q^L}$, and this is additionally multiplied by a tower of 2's of height roughly L.

To the best of our knowledge, prior work on this question focuses exclusively on the $q = 2$ case. For example, in the case of $L = 2$ (i.e., unique-decoding) we have $p_*(2, 1, 2) = 1/4$, and work by Levenshtein [20] shows that a code construction based on Hadamard matrices corrects a $1/4 + \varepsilon$ fraction of errors and has size $1/(4\varepsilon) + O(1)$. A particularly relevant prior work is due to Alon, Bukh and Polyanskiy [1]. Herein the authors consider this question for the special case of $q = 2$ (and thus, necessarily, only for list-decoding). In particular, they show that when L is even if $p = p_*(2, 1, L) + \varepsilon$ then such a (p, L)-list-decodable code $\mathcal{C} \subseteq [2]^n$ has size at most $O_L(1/\varepsilon)$, and moreover provide a construction of such a code with size $\Omega_L(1/\varepsilon)$.[3] They observe some interesting behaviour in the case of odd L; in particular, the maximum size of a $(p_*(2, 1, 3) + \varepsilon, 3)$-list-decodable code is $\Theta(1/\varepsilon^{3/2})$.[4]

Our motivations for this investigation are three-fold. Firstly, the zero-rate regime offers combinatorial challenges and interesting behaviours that we uncover in this work. Secondly, many codes that find applications in other areas of theoretical computer in fact have subconstant rate. Lastly, the zero-rate regime appears much more tractable than the positive rate regime – indeed, we can obtain tight upper and lower bounds on the size of a code, as we will soon see. It would be interesting to determine to what extent such techniques could be useful for understanding the positive rate regime as well.

[2] Note that setting $\ell = 1$ recovers the standard q-ary entropy function, which itself reduces to the binary entropy function upon setting $q = 2$.

[3] Note that for the special case of $q = 2$, the zero-rate threshold for list-decoding had already been established by Blinovsky [4].

[4] This argument in fact shows a flaw in an earlier claimed proof of Blinovsky that claimed such codes have size $O_L(1/\varepsilon)$ for all $L \in \mathbb{N}$.

1.1 Our results

Our main result in this work is a tight bound on the size of a (p, ℓ, L)-list-recoverable code over an alphabet of size $q \geqslant 3$ when $p > p_*(q, \ell, L)$. The main technical challenge is to compute the following upper bound on the size of such a code.

▶ **Theorem 1.** *Let $q, \ell, L \in \mathbb{N}$ with $q \geqslant 3$. $\ell < q$ and $L > \ell$ be fixed constants. Let $\varepsilon > 0$ and put $p = p_*(q, \ell, L) + \varepsilon$. Suppose $\mathcal{C} \subseteq [q]^n$ is (p, ℓ, L)-list-recoverable. Then $|\mathcal{C}| \leqslant O_{q,\ell,L}(1/\varepsilon)$.*

We complement the above negative result with the following code construction, showing the upper bound is tight.

▶ **Theorem 2.** *Let $q, \ell, L \in \mathbb{N}$ with $q \geqslant 3$ and $\ell < q$ be fixed constants. Let $\varepsilon > 0$ and put $p = p_*(q, \ell, L) + \varepsilon$. There exists a (p, ℓ, L)-list-recoverable code $\mathcal{C} \subseteq [q]^n$ such that $|\mathcal{C}| \geqslant \Omega_{q,\ell,L}(1/\varepsilon)$.*

We emphasize that in the above theorems the implied constants may depend on q, ℓ and L.

Note that our results explicitly *exclude* the case of $q = 2$. As [1] prove, the binary alphabet behaves in subtle ways: the bound on the code size depends on the parity of L. Intriguingly, our work demonstrates that such behaviour does not arise over larger alphabets.

1.2 Technical Overview

The double-counting argument. Since our focus is on zero-rate list-decodable/-recoverable codes, it helps to first review the proof of the zero-rate threshold $p_*(q, \ell, L)$. A lower bound can be easily obtained by a random construction that attains a positive rate for any $p \leqslant p_*(q, \ell, L) - \varepsilon$. For the upper bound, let us first consider the list-decoding case, i.e., $\ell = 1$. The proof in [5, 6, 25], at a high-level, proceeds via a double-counting argument.[5] For any (p, ℓ, L)-list-decodable code $\mathcal{C} \subset [q]^n$, the proof aims to upper and lower bound the *radius* of a list averaged over the choice of the list from $\mathcal{C}$:

$$\frac{1}{M^L} \sum_{(\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L) \in \mathcal{C}^L} \mathrm{rad}_{\mathrm{H}}(\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L). \tag{1}$$

Comparing the bounds produces an upper bound on $|\mathcal{C}|$. Here $\mathrm{rad}_{\mathrm{H}}(\cdot)$, known as the Chebyshev radius of a list, is the relative radius of the smallest Hamming ball containing all codewords in the list. A lower bound on Equation (1) essentially follows from list-decodability of $\mathcal{C}$. Indeed, each term (corresponding to lists consisting of distinct codewords) is lower bounded by p, otherwise a list that fits into a ball of radius at most np is found, violating list-decodability of $\mathcal{C}$. Therefore Equation (1) is at least $p - o(1)$, where $o(1)$ is to account for lists with not-all-distinct codewords.

On the other hand, it is much more tricky to upper bound Equation (1) as, in general, $\mathrm{rad}_{\mathrm{H}}$ admits no analytically closed form and can only be computed by solving a min-max problem. Previous proofs [25] then first extracts a subcode $\mathcal{C}'$ with highly-regular list structures via the hypergraph Ramsey's theorem. This allows one to assert that all lists have essentially the same radius and all codewords in each list have essentially the same distance to the center of the list. As a result, the min-max expression is "linearized" and Equation (1) can be upper bounded when restricted to $\mathcal{C}'$. The downside is that the Ramsey reduction step is rather lossy for code size.

[5] A characterization of $p_*(q, 1, L)$ was announced in [5, 6] whose proof was flawed. The work [25] filled in the gaps therein and characterized $p_*(q, \ell, L)$ for general ℓ.

Weighted average radius. The effect of the Ramsey reduction, put formally, is to enforce the *average radius*:

$$\overline{\mathrm{rad}}_{\mathrm{H}}(\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L) \coloneqq \frac{1}{n} \min_{\boldsymbol{r} \in \{0,1\}^n} \frac{1}{L} \sum_{i=1}^{L} d_{\mathrm{H}}(\boldsymbol{c}_i, \boldsymbol{r}) \tag{2}$$

of every list in the subcode to be approximately equal. To extract the regularity structures in lists without resorting to extremal bounds from Ramsey theory, [1] introduced the notion of *weighted average radius* which "linearizes" the Chebyshev radius in a *weighted* manner:

$$\overline{\mathrm{rad}}_{\omega}(\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L) \coloneqq \frac{1}{n} \min_{\boldsymbol{r} \in \{0,1\}^n} \sum_{i=1}^{L} \omega(i) d_{\mathrm{H}}(\boldsymbol{c}_i, \boldsymbol{r})$$

where ω is a distribution on L elements. For any weighting ω, $\overline{\mathrm{rad}}_{\omega}$ of lists from the code forms a suite of succinct statistics of the list distribution. It turns out $\overline{\mathrm{rad}}_{U_L} = \overline{\mathrm{rad}}$ (where U_L denotes the uniform distribution on $[L]$) is maximal under all ω. Recall that the double-counting argument suggests that in an optimal zero-rate code, the behaviour of the ensemble average of rad is essentially captured by that of $\overline{\mathrm{rad}}$. In particular, list-decodability ensures that $\overline{\mathrm{rad}}$ of *most* lists should be large. However, not too many lists in an optimal code are expected to have large $\overline{\mathrm{rad}}_{\omega}$ for any $\omega \neq U_L$. [1] then managed to quantify the gap between $\overline{\mathrm{rad}} = \overline{\mathrm{rad}}_{U_L}$ and $\overline{\mathrm{rad}}_{\omega}$ (with $\omega \neq U_L$), which yields improved (and sometimes optimal) size-radius trade-off of zero-rate codes.

Generalization to q-ary list-decoding. Our major technical contribution is in extrapolating the above ideas to list-recovery. The challenge lies particularly in defining a proper notion of weighted average radius and proving its properties. Our definition relies crucially on an embedding φ from $[q]$ to the simplex in $\mathbb{R}^q$ and relaxes the center $\boldsymbol{r}$ of the list to be a fractional vector. Specifically, denoting by $\Delta = \left\{\boldsymbol{a} \in \mathbb{R}^q_{\geqslant 0} : \sum_{i=1}^{q} a_i = 1\right\}$ the simplex in $\mathbb{R}^q$ and $\partial\Delta = \{\boldsymbol{e}_1, \cdots, \boldsymbol{e}_q\}$ its vertices (i.e., the standard basis of $\mathbb{R}^q$), we let the embedding φ map each symbol $x \in [q]$ to the standard basis vector $\boldsymbol{e}_x \in \partial\Delta$. Denoting by $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L \in (\partial\Delta)^n$ the (element-wise) images of a list $\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L \in [q]^n$, we define the *weighted average radius* of $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L$ as:

$$\overline{\mathrm{rad}}_{\omega}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L) = \frac{1}{n} \min_{\boldsymbol{y} \in \Delta^n} \frac{1}{2} \mathop{\mathbb{E}}_{i \sim \omega} \left[\|\boldsymbol{x}_i - \boldsymbol{y}\|_1\right], \tag{3}$$

where ω is any distribution on $[L]$.

The notriviality and significance of the above notion, especially the embedding used therein, is three-fold.

- First, as the weighting ω varies, $\overline{\mathrm{rad}}_{\omega}$ serves as a bridge between the standard average radius in Equation (2) and the Chebyshev radius. Indeed, $\omega = U_L$ recovers the former, and the maximum $\overline{\mathrm{rad}}_{\omega}$ over ω recovers the latter. However, we caution that the second statement does not hold without the embedding since the Hamming distance between q-ary symbols per se is not and cannot be interpolated by a convex function, which makes the minimax theorem inapplicable. Fortunately, our embedding affinely extend the q-ary Hamming distance to the simplex therefore brings back the applicability of the minimax theorem and connects $\max_{\omega} \overline{\mathrm{rad}}_{\omega}$ to rad.
- Second, our definition in Equation (3) allows $\boldsymbol{y}$ to take any value on the simplex, instead of only its vertices, i.e., the image of $[q]$ under φ. Though embedding naively to the hypercube $[0,1]^q$ seems convenient, upon solving the expression with fractional $\boldsymbol{y}$ one

does not necessarily obtain a notion that is guaranteed to closely approximate the original version with integral $\boldsymbol{y}$. In contrast, using linear programming duality, we show that our embedding yields relaxed notion of radius which closely approximates the actual Chebyshev radius. Indeed, upon rounding the fractional center $\boldsymbol{y}$ and taking its pre-image under φ, our results guarantee that the resulting radius must have negligible difference from the Chebyshev radius. Precisely speaking, we want to find a vector $\boldsymbol{y} = (y(i,j))_{[n]\times[q]} \in \Delta$ close to the L images of the codewords $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L$ by linear programming. Meanwhile, we want $\boldsymbol{y}(i) := (y(i,1), \ldots, y(i,q))$ to belong to $\partial\Delta$ so that we can find a preimage of $\boldsymbol{y}(i)$ in $[q]$. Since $\boldsymbol{y}(i) \in \Delta$, the components in $\boldsymbol{y}(i)$ are subject to $\sum_{j=1}^{q} y(i,j) = 1$. This implies that at least one component of $\boldsymbol{y}(i)$ is nonzero. The basic feasible solution guarantees that there exists a feasible solution such that most of $y(i,j)$ are 0. Combining with the fact $\sum_{j=1}^{q} y(i,j) = 1$ forces $(y(i,1), \ldots, y(i,q)) \in \partial\Delta$ for almost all $i \in [n]$. Thus, we obtain a negligible loss in the conversion between Hamming distance and Euclidean distance.

- Finally, under the embedding φ, the weighted average radius $\overline{\mathrm{rad}}_\omega$ still retains the appealing feature that the minimization can be analytically solved, therefore giving rise to an explicit expression (see Equation (14)) which greatly facilitates our analysis.

We then show, via techniques deviating from those in [1], three key properties that are required by the subsequent arguments.

1. For any fixed distribution P, if entries of codewords in the list are generated i.i.d. using P, then

$$f(P,\omega) := \mathop{\mathbb{E}}_{(X_1,\cdots,X_L)\sim P^{\otimes L}} \left[1 - \max_{x\in[q]} \sum_{\substack{i\in[L] \\ X_i = x}} \omega(i) \right]$$

   is maximized when $\omega = U_L$. Moreover, the equality holds if and only if $\omega = U_L$ for $q \geqslant 3$ and any L. Our approach is different from [1] as we can not explicitly represent function $f(P,\omega)$.
2. Furthermore, if entries of codewords in the list are generated i.i.d. using a certain P, then $f(P, U_L)$ is upper bounded by $f(P_{q,p}, U_L)$ with $P_{q,p} = (\frac{1-p}{q}, \ldots, \frac{1-p}{q}, p)$ and $p = \max_{i\in[q]} P(i)$. This follows from the Schur convexity property proved in [25].
3. Finally, denoting by P_i the distribution of the i-th components of codewords in code $\mathcal{C}$, Schur convexity promises $f(P_i, U_L) \leqslant f(P_{q,p_i}, U_L)$. In [25], it is proved that $f(P_{q,p}, U_L)$ is convex for $p \in [1/q, 1]$. Thus, we can conclude that

$$\frac{1}{n} \sum_{i\in[n]} f(P_i, U_L) \leqslant f(P_{q,p}, U_L)$$

   with $p = \frac{1}{n}\sum_{i\in[n]} p_i$.

   The remaining part of our proof is similar to [1]. We show that a code $\mathcal{C}$ either has radius

$$\mathrm{rad}(\mathcal{C}) = \frac{1}{n} \min_{\boldsymbol{x}\in[q]^n} \max_{\boldsymbol{c}\in\mathcal{C}} d_{\mathrm{H}}(\boldsymbol{c},\boldsymbol{x}) \leqslant 1 - \frac{1}{q} - \delta$$

or most of L-tuples with distinct codewords in $\mathcal{C}$ are distributed close to uniform,. For the former case, we use the convexity property to show that the list-decodability of $\mathcal{C}$ can not exceed $f(U_q, U_L) = p_*(q, L)$ by much. For the latter case, since most of L-tuples of distinct codewords in $\mathcal{C}^L$ looks uniformly at random, we can show that the list-decodablilty of $\mathcal{C}$ is very close to that of random codes which is $f(U_q, U_L)$.

Generalization to list-recovery. For list-recovery, i.e., $\ell > 1$, we find an embedding φ_ℓ that maps each element in $[q]$ to a superposition of ℓ vertices of the simplex in $\mathbb{R}^q$, i.e., we map the element in $[q]$ to a vector space $[0,1]^{\mathcal{X}}$ where $\mathcal{X} = \binom{[q]}{\ell}$ is the collection of all ℓ-subsets in $[q]$. Concretely, we define $\varphi_\ell(i) := \sum_{A \in \mathcal{X}, i \in A} \boldsymbol{e}_A$ where $(\boldsymbol{e}_A)_{A \in \mathcal{X}}$ is a standard basis of $\mathbb{R}^{\mathcal{X}}$. The intuition behind this map is that if $i \in X$, we have $\|\varphi_\ell(i) - \boldsymbol{e}_X\|_1 = \binom{q}{\ell} - 1$ and otherwise $\|\varphi_\ell(i) - \boldsymbol{e}_X\|_1 = \binom{q}{\ell} + 1$. Similar to the list decoding, given L codewords in $[q]^n$, we obtain L vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_L$ under the map φ_ℓ. Our goal is to find a vector $\boldsymbol{y} = (y(i,A))_{[n] \times \mathcal{X}}$ close to these L vectors subject to the constraint that $\sum_{A \in \mathcal{X}} y(i,A) = 1$ for any $i \in [n]$. This constraint combined with the basic feasible solution argument forces that for almost all $i \in [n]$, $(y(i,A))_{A \in \mathcal{X}}$ is of the form $\boldsymbol{e}_X$. For such i, we can find an ℓ-subset $X \in \mathcal{X}$ preserving the distance, i.e.,

$$d_{\mathrm{LR}}(i, X) = \mathbb{1}\{i \notin X\} = \frac{1}{2}\left(\|\varphi_\ell(i) - \boldsymbol{e}_X\|_1 - \binom{q}{\ell} + 1\right).$$

Besides the linear programming relaxation, further adjustments for the proof of properties analogous to Items 1–3 above are required.

Code construction. As alluded to before, a code that saturates the optimal size-radius trade-off should essentially saturate both the upper and lower bounds on the quantity

$$\frac{1}{M^L} \sum_{(\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L) \in \mathcal{C}^L} \overline{\mathrm{rad}}_{\mathrm{H}}(\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L)$$

considered in the double-counting argument. Indeed, our impossibility result implies that any optimal zero-rate code must contain a large fraction of random-like L-tuples $(\boldsymbol{c}_1, \dots, \boldsymbol{c}_L)$, i.e., for every $\boldsymbol{u} \in [q]^L$

$$\sum_{i=1}^{n} \mathbb{1}\{(\boldsymbol{c}_1(i), \dots, \boldsymbol{c}_L(i)) = \boldsymbol{u}\} \approx \frac{n}{q^L} \tag{4}$$

where $\boldsymbol{c}_j = (\boldsymbol{c}_j(1), \dots, \boldsymbol{c}_j(n)) \in [q]^n$. To match such an impossibility result, an optimal construction should contain as many such L-tuples as possible. A simplex-like code then becomes a natural candidate. This is a natural extension of the construction in [1] to larger alphabet. An $M \times n$ codebook $\mathcal{C}$ consisting of M codewords each of length n is constructed by putting as columns all possible distinct length-M vectors that contains identical numbers of $1, 2, \cdots, q$. It is not hard to see by symmetry that (4) becomes equality for every L-tuple with distinct codewords in $\mathcal{C}$. Thus, $\mathcal{C}$ is the most regular code.

We also remark that, unlike for positive-rate codes, the prototypical random construction (with expurgation) does not lead to favorable size-radius trade-off since the deviation of random sampling is comparatively too large in the zero-rate regime. In contrast, the simplex code is deterministically regular and has no deviation.

1.3 Organization

The remainder is organized as follows. First, Section 2 provides the necessary notations and definitions, together with some preliminary results which will be useful in the subsequent arguments. Sections 3.1–3.4 contain our argument establishing Theorem 1 for list-decoding (i.e. the case $\ell = 1$); in Section 4 we elucidate the changes that need to be made to establish the theorem for general ℓ. Next, Section 5 provides the code construction establishing Theorem 2.

2 Preliminaries

Firstly, for convenience of the reader we begin by summarizing the notation that we use. This is particularly relevent as we will often be in situations where we need multiple indexes for, e.g., lists of vectors where each coordinate lies in a probability simplex, so the reader is encouraged to refer to this table whenever it is unclear what is intended.

Table 1 Notation for list-decoding.

English letter in boldface	$[q]^n$-valued vector
Greek letter in boldface	$\Delta([q])$-valued vector
$\Delta := \Delta([q])$	Simplex in $[0,1]^q$, i,e., $\Delta = \{(x_1, \dots, x_q) \in [0,1]^q : \sum_{i=1}^q x_i = 1\}$
$\partial\Delta$	Set of vertices of Δ
$\boldsymbol{e}_x \in \partial\Delta$	The image of $x \in [q]$ under φ, i.e., the x-th vertex of Δ
$\boldsymbol{c}_i \in [q]^n$	The i-th codeword in a list
$\boldsymbol{x}_i \in \Delta^n$	Image of $\boldsymbol{c}_i$ under φ (applied component-wise)
$\boldsymbol{y} \in \Delta^n$	Relaxed center of a list
$\boldsymbol{x}(j) \in \partial\Delta, \boldsymbol{y}(j) \in \Delta$	The j-th block (of length q) in $\boldsymbol{x} \in (\partial\Delta)^n, \boldsymbol{y} \in \Delta^n$, respectively
$x(j,k) \in \{0,1\}, y(j,k) \in [0,1]$	The (j,k)-th element of $\boldsymbol{x} \in (\partial\Delta)^n, \boldsymbol{y} \in \Delta^n$, respectively
$\mathrm{rad}_{\mathrm{H}}$	(Standard) Chebyshev radius
rad	Relaxed Chebyshev radius
$\overline{\mathrm{rad}}$	Average radius
$\overline{\mathrm{rad}}_{\omega}$	Average radius weighted by $\omega \in \Delta([L])$
$f(P,\omega)$	Expected average radius (weighted by ω) of P-distributed symbols
$(X_1, \cdots, X_L) \sim P^{\otimes L}$	A list of i.i.d. P-distributed symbols
U_k	Uniform distribution on $[k]$

For a finite set S and an integer $0 \leqslant k \leqslant |S|$, we denote $\binom{S}{k} := \{T \subset S : |T| = k\}$. Let $[q] = \{1, \dots, q\}$.

2.1 List-Decoding

Fix $q \in \mathbb{Z}_{\geqslant 3}$ and $L \in \mathbb{Z}_{\geqslant 2}$. Let $d_{\mathrm{H}}(\boldsymbol{c}, \boldsymbol{r})$ denote the *Hamming distance* between $\boldsymbol{c}, \boldsymbol{r} \in [q]^n$, i.e., the number of coordinates on which the strings differ. For $t \in [0,n]$, let $\mathcal{B}_{\mathrm{H}}(\boldsymbol{y}, t) := \{\boldsymbol{c} \in [q]^n : d_{\mathrm{H}}(\boldsymbol{c}, \boldsymbol{y}) \leqslant t\}$ denote the *Hamming ball* centered around $\boldsymbol{y}$ of radius $\lfloor t \rfloor$.

▶ **Definition 3** (List-decodable code). *Let $p \in [0,1]$. A code $\mathcal{C} \subseteq [q]^n$ is $(p, L)_q$-*list-decodable *if for any $\boldsymbol{y} \in [q]^n$,*

$$|\mathcal{C} \cap \mathcal{B}_{\mathrm{H}}(\boldsymbol{y}, np)| \leqslant L - 1.$$

In [25] the zero-rate regime for list-decoding was derived, which is the supremum over $p \in [0,1]$ for which $(p - \varepsilon, L)_q$-list-decodable codes of positive rate exist for all $\varepsilon > 0$. This value was shown to be

$$p_*(q, L) = 1 - \frac{1}{L} \mathop{\mathbb{E}}_{(X_1, \cdots, X_L) \sim U_q^{\otimes L}} [\mathsf{pl}(X_1, \cdots, X_L)], \tag{5}$$

where the function pl outputs the number of times the most popular symbol appears. In [25] it is shown that $(p_*(q, L) + \varepsilon, L)$-list-decodable codes have size $O_{\varepsilon,q,L}(1)$, i.e., some constant independent of n. Our target in this work is to show that the correct dependence on ε is $O_{q,L}(1/\varepsilon)$, except for the case of $q = 2$ with odd L.

A "dual" definition of list-decodability is proffered by the Chebyshev radius.

▶ **Definition 4** (Chebyshev radius). *The* Chebyshev radius *of a list of distinct vectors* $\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L \in [q]^n$ *is defined as*

$$\mathrm{rad}_{\mathrm{H}}(\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L) \coloneqq \frac{1}{n} \min_{\boldsymbol{r} \in [q]^n} \max_{i \in [L]} d_{\mathrm{H}}(\boldsymbol{c}_i, \boldsymbol{r}).$$

Observe that a code $\mathcal{C} \subseteq [q]^n$ is (p, L)-list-decodable if and only if

$$\min\{\mathrm{rad}_{\mathrm{H}}(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_L) : \boldsymbol{c}_1, \ldots, \boldsymbol{c}_L \in \mathcal{C} \text{ distinct}\} > p \ . \tag{6}$$

In particular, to show a code fails to be list-decodable, it suffices to upper bound the Chebyshev radius of L distinct codewords.

Recall that our main target is an upper bound on the size of list-decodable/-recoverable codes (in the zero-rate regime). A natural approach is to derive from Equation (6) the desired bound on the code; however, this quantity is quite difficult to work with directly. We therefore work with a *relaxed* version, which we now introduce.

We require the following definitions. Let us embed $[q]^n$ into the simplex $\Delta([q])$ via the following map φ:

$$\begin{array}{rccc} \varphi\colon & [q] & \to & \Delta([q]) \\ & x & \mapsto & \boldsymbol{e}_x \end{array} \tag{7}$$

where $\boldsymbol{e}_x$ is the q-dimensional vector with a 1 in the x-th location and 0 everywhere else. Denote by $\Delta = \Delta([q])$ the simplex and $\partial\Delta = \{\boldsymbol{e}_1, \cdots, \boldsymbol{e}_q\}$ its vertices. For $\boldsymbol{\chi} = \boldsymbol{e}_x \in \partial\Delta$ and $\boldsymbol{\eta} \in \Delta$, let

$$d(\boldsymbol{\chi}, \boldsymbol{\eta}) \coloneqq \frac{1}{2} \|\boldsymbol{\chi} - \boldsymbol{\eta}\|_1 = \frac{1}{2} \left(1 - \boldsymbol{\eta}(x) + \sum_{x' \in [q] \setminus \{x\}} \boldsymbol{\eta}(x') \right). \tag{8}$$

Note that if $\boldsymbol{\eta} = \boldsymbol{e}_y \in \partial\Delta$, then

$$d(\boldsymbol{\chi}, \boldsymbol{\eta}) = d_{\mathrm{H}}(x, y).$$

From now on we will only work with Δ^n-valued vectors and will still denote such length-qn vectors by boldface letters, abusing the notation. For $\boldsymbol{y} \in \Delta^n$, we use $y(j,k) \in [0,1]$ to denote its (j,k)-th element and use $\boldsymbol{y}(j) = (y(j,1), \ldots, y(j,q)) \in \Delta$ to denote its j-th block of size q. For $\boldsymbol{c} \in [q]^n$, we use $\boldsymbol{c}(j) \in [q]$ to denote its j-th element.

For $\boldsymbol{x} \in (\partial\Delta)^n$ and $\boldsymbol{y} \in \Delta^n$, the definition of $d(\cdot,\cdot)$ can be extended to length-qn vectors in the natural way. Specifically,

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sum_{j=1}^{n} d(\boldsymbol{x}(j), \boldsymbol{y}(j)). \tag{9}$$

We may now define the *relaxed Chebyshev radius*.

▶ **Definition 5.** *The* relaxed Chebyshev radius *of a list of distinct vectors* $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L \in (\partial\Delta)^n$ *is*

$$\mathrm{rad}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L) \coloneqq \frac{1}{n} \min_{\boldsymbol{y} \in \Delta^n} \max_{i \in [L]} d(\boldsymbol{x}_i, \boldsymbol{y}). \tag{10}$$

Observe that

$$\mathrm{rad}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L) \leqslant \mathrm{rad}_{\mathrm{H}}(\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L). \tag{11}$$

where $\varphi(\boldsymbol{c}_i) = \boldsymbol{x}_i$ (here we extend the definition of φ to length-n inputs in a similar way as in Equation (9)). This justifies the "relaxation" terminology.

As a last piece of terminology, we define the radius of a code.

▶ **Definition 6.** *For any code* $\mathcal{C} \in [q]^n$*, we define the* Chebyshev radius *of* $\mathcal{C}$ *as*

$$\mathrm{rad}(\mathcal{C}) = \frac{1}{n} \min_{\boldsymbol{x} \in [q]^n} \max_{\boldsymbol{c} \in \mathcal{C}} d_{\mathrm{H}}(\boldsymbol{c}, \boldsymbol{x}).$$

3 Zero-Rate List-Decoding

3.1 Linear Programming Relaxation

We have shown in Equation (11) that rad is smaller than $\mathrm{rad}_{\mathrm{H}}$. Conversely, Lemma 7 below establishes that rad and $\mathrm{rad}_{\mathrm{H}}$ do not differ much. That is, for any list, if a center $\boldsymbol{y} \in \Delta^n$ achieves a relaxed radius t, then there must exist $\boldsymbol{r} \in [q]^n$ attaining approximately the same t for sufficiently large n.

▶ **Lemma 7** (rad is close to $\mathrm{rad}_{\mathrm{H}}$)**.** *Let $\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L \in [q]^n$. Denote by $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L \in (\partial\Delta)^n$ the images of $\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L$ under the embedding φ. Then*

$$\mathrm{rad}_{\mathrm{H}}(\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L) \leqslant \mathrm{rad}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L) + \frac{L}{n}.$$

Proof. Suppose $\mathrm{rad}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L) = t$. Then there exists $\boldsymbol{y} \in \Delta^n$ such that for every $i \in [L]$,

$$d(\boldsymbol{x}_i, \boldsymbol{y}) = \frac{1}{2}\sum_{j=1}^{n}\left(1 - y(j, c_i(j)) + \sum_{x \in [q]\setminus\{c_i(j)\}} y(j,x)\right) \leqslant t,$$

where the first equality is by Equations (8) and (9). That is, the following polytope is nonempty:

$$\begin{aligned}&\{\boldsymbol{y} \in \Delta^n : \forall i \in [L],\ d(\boldsymbol{x}_i, \boldsymbol{y}) \leqslant t\}\\ &= \left\{ (y(j,k))_{(j,k)\in[n]\times[q]} : \begin{array}{l} \forall (j,k) \in [n]\times[q],\ y(j,k) \geqslant 0,\\ \forall j \in [n],\ \sum_{k=1}^{q} y(j,k) = 1,\\ \forall i \in [L],\ \frac{1}{2}\sum_{j=1}^{n}\left(1 - y(j,\boldsymbol{c}_i(j)) + \sum_{x\in[q]\setminus\{\boldsymbol{c}_i(j)\}} y(j,x)\right) \leqslant t \end{array} \right\}.\end{aligned}$$

Equivalently, the following linear program (LP) is feasible:

$$\begin{array}{ll} \max\limits_{(y(j,k))_{(j,k)\in[n]\times[q]}} & 0\\ \text{s.t.} & \forall (j,k) \in [n]\times[q],\ y(j,k) \geqslant 0,\\ & \forall j \in [n],\ \sum_{k=1}^{q} y(j,k) = 1,\\ & \forall i \in [L],\ \frac{1}{2}\sum_{j=1}^{n}\left(1 - y(j,\boldsymbol{c}_i(j)) + \sum_{x\in[q]\setminus\{\boldsymbol{c}_i(j)\}} y(j,x)\right) \leqslant t. \end{array}$$

Since the equality $\langle \boldsymbol{a}, \boldsymbol{y}\rangle \leqslant b$ is equivalent to the equality $\langle \boldsymbol{a}, \boldsymbol{y}\rangle + z = b, z \geqslant 0$, the above LP can be written in equational form:

$$\begin{array}{ll} \max\limits_{(y(j,k))_{(j,k)\in[n]\times[q]}, (z(i))_{i\in[L]}} & 0\\ \text{s.t.} & \forall (j,k) \in [n]\times[q],\ y(j,k) \geqslant 0,\\ & \forall i \in [L],\ z(i) \geqslant 0,\\ & \forall j \in [n],\ \sum\limits_{k=1}^{q} y(j,k) = 1,\\ & \forall i \in [L],\ \frac{1}{2}\sum\limits_{j=1}^{n}\left(1 - y(j,\boldsymbol{c}_i(j)) + \sum\limits_{x\in[q]\setminus\{\boldsymbol{c}_i(j)\}} y(j,x)\right) + z(i) \leqslant t, \end{array} \tag{12}$$

or more compactly in matrix form:

$$\begin{array}{ll} \max\limits_{\boldsymbol{y}\in\mathbb{R}^{nq}, \boldsymbol{z}\in\mathbb{R}^L} & 0\\ \text{s.t.} & \begin{bmatrix} A & I_L \\ B & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} t\mathbf{1}_L \\ \mathbf{1}_n \end{bmatrix},\\ & \boldsymbol{y}, \boldsymbol{z} \geqslant \mathbf{0}. \end{array}$$

Here $A \in \mathbb{R}^{L\times(nq)}$ and $B \in \mathbb{R}^{n\times(nq)}$ encode respectively the fourth and third constraints in Equation (12), and $I_L \in \mathbb{R}^{L\times L}, \mathbf{1}_L \in \mathbb{R}^L$ denote respectively the $L \times L$ identity matrix and the all-one vector of length L. It is clear that $\mathrm{rk}\left(\begin{bmatrix} A & I_L \\ B & 0 \end{bmatrix}\right) \leqslant n+L$. This implies that there exists a feasible solution $\boldsymbol{y}, \boldsymbol{z}$ that has at most $n+L$ nonzeros and thus $\boldsymbol{y} = (y(j,k))_{(j,k)\in[n]\times[q]}$ has at most $n+L$ nonzeros. Indeed, such solutions are known as the *basic feasible solutions*. Note that for every block $j \in [n]$, $\sum_{k=1}^{q} y(j,k) = 1$. This implies that $y(j,1),\dots,y(j,q)$ cannot be simultaneously 0. Moreover, if $q-1$ out of them are 0, the remaining one is forced to be 1. Since there are n blocks in total, by the pigeonhole principle, there are at least $n-L$ choices of $j \in [n]$ such that $\boldsymbol{y}(j) = (y(j,1),\dots,y(j,q)) \in \partial\Delta$. Without loss of generality, we assume that these $n-L$ indices are $1,\dots,n-L$. Let $\boldsymbol{r} \in [q]^n$ be such that $\varphi(\boldsymbol{r}(j)) = \boldsymbol{y}(j)$ for $j = 1,\dots,n-L$ and $r(j)$ is any value in $[q]$ for $j = n-L+1,\dots,n$. Since $d(\boldsymbol{x}_i(j), \boldsymbol{y}(j)) \in [0,1]$ and $d_{\mathrm{H}}(\boldsymbol{c}_i(j), \boldsymbol{r}(j)) \in \{0,1\}$, the difference between $d(\boldsymbol{x}_i, \boldsymbol{y})$ and $d_{\mathrm{H}}(\boldsymbol{c}_i, \boldsymbol{r})$ is at most L. The proof is completed. ◀

We further relax rad by defining the *weighted average radius*. For $\boldsymbol{x}_1,\cdots,\boldsymbol{x}_L \in (\partial\Delta)^n$ and $\omega \in \Delta([L])$, let

$$\overline{\mathrm{rad}}_\omega(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_L) \coloneqq \frac{1}{n}\min_{\boldsymbol{y}\in\Delta^n} \mathop{\mathbb{E}}_{i\sim\omega}\left[d(\boldsymbol{x}_i,\boldsymbol{y})\right] = \frac{1}{n}\min_{\boldsymbol{y}\in\Delta^n}\sum_{i\in[L]}\omega(i)d(\boldsymbol{x}_i,\boldsymbol{y}).$$

In words, weighted average radius is obtained by replacing the maximization over $i \in [L]$ in the definition of relaxed radius (see Equation (10)) with an average with respect to a distribution ω.

Since the objective of the minimization is separable, one can minimize over each $y(j)$ individually and obtain an alternative expression. Suppose $\boldsymbol{x}_1,\cdots,\boldsymbol{x}_L \in (\partial\Delta)^n$ are the images of $\boldsymbol{c}_1,\cdots,\boldsymbol{c}_L \in [q]^n$ under the embedding φ. Then

$$\begin{aligned}
\overline{\mathrm{rad}}_\omega(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_L) &= \frac{1}{n}\min_{\boldsymbol{y}\in\Delta^n}\sum_{i\in[L]}\omega(i)d(\boldsymbol{x}_i,\boldsymbol{y}) \\
&= \frac{1}{2n}\min_{(\boldsymbol{y}_1,\cdots,\boldsymbol{y}_n)\in\Delta^n}\sum_{j=1}^{n}\left[\sum_{i\in[L]}\omega(i)\left(1-\boldsymbol{y}_j(\boldsymbol{c}_i(j))+\sum_{x\in[q]\setminus\{\boldsymbol{c}_i(j)\}}\boldsymbol{y}_j(x)\right)\right] \\
&= \frac{1}{2n}\sum_{j=1}^{n}\min_{\boldsymbol{y}_j\in\Delta}\left[\sum_{i\in[L]}\omega(i)\left(1-2\boldsymbol{y}_j(\boldsymbol{c}_i(j))+\sum_{x\in[q]}\boldsymbol{y}_j(x)\right)\right] \qquad (13) \\
&= \frac{1}{2n}\sum_{j=1}^{n}\min_{\boldsymbol{y}_j\in\Delta}\left[\sum_{i\in[L]}\omega(i)\left(2-2\boldsymbol{y}_j(\boldsymbol{c}_i(j))\right)\right] \\
&= \frac{1}{n}\sum_{j=1}^{n}\min_{\boldsymbol{y}_j\in\Delta}\left[1-\sum_{i\in[L]}\omega(i)\boldsymbol{y}_j(\boldsymbol{c}_i(j))\right] = 1-\frac{1}{n}\sum_{j=1}^{n}\max_{x\in[q]}\sum_{\substack{i\in[L]\\ \boldsymbol{c}_i(j)=x}}\omega(i). \qquad (14)
\end{aligned}$$

Equation (13) holds since the objective in brackets only depends on $\boldsymbol{y}_j$, not on other $(\boldsymbol{y}_{j'})_{j'\in[n]\setminus\{j\}}$. To see Equation (14), we note that a maximizer $\boldsymbol{y}^* \in \Delta$ to the following problem $\max_{\boldsymbol{y}\in\Delta}\sum_{i\in[L]}\omega(i)y(x_i)$, where $\omega \in \Delta([L])$ and $(x_1,\cdots,x_L) \in [q]^L$ are fixed, is given by $\boldsymbol{y}^* = \boldsymbol{e}_{x^*}$ where $x^* \in [q]$ satisfies $x^* \in \mathrm{argmax}_{x\in[q]}\sum_{i\in[L]}\omega(i)\mathbb{1}\{x_i = x\}$.

Obviously, by definition, for any $\boldsymbol{x}_1,\cdots,\boldsymbol{x}_L \in (\partial\Delta)^n$ and $\omega \in \Delta([L])$, $\overline{\mathrm{rad}}_\omega(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_L) \leqslant \mathrm{rad}(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_L)$. In fact, the following lemma shows that rad is equal to the maximum $\overline{\mathrm{rad}}_\omega$ over ω.

▶ **Lemma 8** (rad equals maximum $\overline{\mathrm{rad}}_\omega$)**.** *For any* $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L \in (\partial\Delta)^n$,

$$\mathrm{rad}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L) = \max_{\omega \in \Delta([L])} \overline{\mathrm{rad}}_\omega(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L).$$

Proof. Note that

$$\mathrm{rad}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L) \coloneqq \frac{1}{n} \min_{\boldsymbol{y} \in \Delta^n} \max_{i \in [L]} d(\boldsymbol{x}_i, \boldsymbol{y}) = \frac{1}{n} \min_{\boldsymbol{y} \in \Delta^n} \max_{\omega \in \Delta([L])} \sum_{i \in [L]} \omega(i) d(\boldsymbol{x}_i, \boldsymbol{y}),$$

since the inner maximum is anyway achieved by a singleton distribution. Note also that the objective function

$$\sum_{i \in [L]} \omega(i) d(\boldsymbol{x}_i, \boldsymbol{y}) = \frac{1}{2} \sum_{i \in [L]} \omega(i) \sum_{j=1}^{n} \left(1 - y(j, \boldsymbol{c}_i(j)) + \sum_{x \in [q] \setminus \{\boldsymbol{c}_i(j)\}} y(j, x) \right)$$

is affine in ω and linear in $\boldsymbol{y}$. Therefore, von Neumann's minimax theorem allows us to interchange min and max and obtain

$$\mathrm{rad}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L) = \frac{1}{n} \max_{\omega \in \Delta([L])} \min_{\boldsymbol{y} \in \Delta^n} \sum_{i \in [L]} \omega(i) d(\boldsymbol{x}_i, \boldsymbol{y}) = \max_{\omega \in \Delta([L])} \overline{\mathrm{rad}}_\omega(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L),$$

as claimed by the lemma. ◀

In fact, we can say something stronger: it is not necessary to maximize over the entire (uncountable) probability simplex $\Delta([L])$. Instead, we can extract a finite subset $\Omega_L \subset \Delta([L])$ and maximize over this set to recover rad. The following lemma is analogous to [1, Lemma 6].

▶ **Lemma 9** (rad is achieved by finitely many ω)**.** *For every L, there exists a finite set of probability measures $\Omega_L \subseteq \Delta([L])$ such that*

$$\mathrm{rad}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L) = \max_{\omega \in \Omega_L} \overline{\mathrm{rad}}_\omega(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L).$$

for all $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L \in \partial\Delta^n$.

Proof. The idea is to view the computation of $\max_{\omega \in \Omega_L} \overline{\mathrm{rad}}_\omega(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L)$ as finding the maximum among some finite set of linear program maxima over some convex polytopes, and then to take Ω_L to be the set of vertices of the defined convex polytopes.

First, we define the convex polytopes based on a (q-ary version of a) signature. For each $\omega \in \Delta([L])$, we define a *signature* for ω which is a function $S_\omega : [q]^L \to [q]$ such that

$$S_\omega(\boldsymbol{u}) \in \operatorname*{argmax}_{x \in [q]} \sum_{i : \boldsymbol{u}(i) = x} \omega(i)$$

for $\boldsymbol{u} \in [q]^L$. Define further the q halfspaces $H_{\boldsymbol{u},x} \coloneqq \{\omega \in \Delta([L]) : \sum_{i:\boldsymbol{u}(i)=x} \omega(i) \geqslant 1/q\}$ for $x \in [q]$. Observe that if $S(\boldsymbol{u}) = x$ where S is a signature for ω then $\omega \in H_{\boldsymbol{u},x}$. Thus, by ranging over the choices for $\boldsymbol{u} \in [q]^L$ and $x \in [q]$ we obtain q^{L+1} halfspaces that partition the $(L-1)$-dimensional space $\Delta([L])$ into at most $\sum_{j \leqslant L-1} \binom{q^{L+1}}{j}$ regions.

For each possible signature $S : [q]^L \to [q]$, let $\Omega_S = \{\omega \in \Delta([L]) \colon S \text{ is a signature for } \omega\}$, and note that Ω_S is a convex polytope. Indeed, it is an intersection over $\boldsymbol{u} \in [q]^L$ of the convex polytopes

$$\{\omega \in \Delta([L]) \colon \exists \text{ signature } \ S_\omega \text{ for } \omega \text{ s.t. } S_\omega(\boldsymbol{u}) = S(\boldsymbol{u})\}$$
$$= \bigcap_{y \in [q] \setminus S(\boldsymbol{u})} \left\{ \omega \in \Delta([L]) \colon \sum_{i : \boldsymbol{u}(i) = S(\boldsymbol{u})} \omega(i) \geqslant \sum_{i : \boldsymbol{u}(i) = y} \omega(i) \right\}$$

where S_ω is a signature for ω. Now, to maximize

$$\overline{\mathrm{rad}}_\omega(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L) = \frac{1}{n} \min_{\boldsymbol{y} \in \Delta^n} \sum_{i \in [L]} \omega(i) d(\boldsymbol{x}_i, \boldsymbol{y})$$

over $\omega \in \Omega_L$, consider the set $T_{\boldsymbol{u}} = \{i \in [n] : (\boldsymbol{x}_1(i), \ldots, \boldsymbol{x}_L(i)) = \boldsymbol{u}\}$ for $\boldsymbol{u} \in [q]^L$ and let $a_{\boldsymbol{u}} = \frac{|T_{\boldsymbol{u}}|}{n}$. We claim it suffices to find the maximum of the following linear function:

$$\sum_{\boldsymbol{u} \in [q]^L} a_{\boldsymbol{u}} y_{\boldsymbol{u}}, \quad \text{s.t.} \quad y_{\boldsymbol{u}} = \sum_{i : \boldsymbol{u}(i) = S(\boldsymbol{u})} \omega(i) \tag{15}$$

over all $\omega \in \Omega_S$. Indeed, by Equation (14), we have

$$\overline{\mathrm{rad}}_\omega(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L) = 1 - \frac{1}{n} \sum_{j=1}^{n} \max_{x \in [q]} \sum_{\substack{i \in [L] \\ c_i(j) = x}} \omega(i).$$

This implies that a maximizer only depends on the index set $T_{\boldsymbol{u}}$, and furthermore that its value is determined by the $a_{\boldsymbol{u}}$'s as in Equation (15).

We can thus take the union of all vertex sets of all polytopes Ω_S for all signatures S. Multiplying this by the $O_{q,L}(1)$ regions defined by all the halfspaces $H_{\boldsymbol{u},x}$ we obtain a finite set of vertices, as desired. ◀

3.2 Properties of $f(P, \omega)$

Now, we consider the expected weighted average radius of a sequence of i.i.d. symbols. Specifically, for $P \in \Delta([q])$ and $\omega \in \Delta([L])$, let

$$f(P, \omega) \coloneqq \mathop{\mathbb{E}}_{(X_1, \cdots, X_L) \sim P^{\otimes L}} \left[\overline{\mathrm{rad}}_\omega(\boldsymbol{e}_{X_1}, \cdots, \boldsymbol{e}_{X_L}) \right].$$

[1] studies $f(P, \omega)$ for $q = 2$ and even L. In this case, one can take advantage of the fact that $P \in \Delta([2])$ may be parametrized by a single real number, and thereby yield a fairly simple expression for $f(P, \omega)$.

Nonetheless, in this subsection, we will show that all properties of $f(P, \omega)$ in [1] holding for $q = 2$ and even L can be generalized to any $q \geqslant 3$ and any L. Let us first provide a more explicit expression for $f(P, \omega)$ using Equation (14):

$$\begin{aligned} f(P, \omega) &\coloneqq \mathop{\mathbb{E}}_{(X_1, \cdots, X_L) \sim P^{\otimes L}} \left[1 - \max_{x \in [q]} \sum_{\substack{i \in [L] \\ X_i = x}} \omega(i) \right] \\ &= 1 - \sum_{(x_1, \cdots, x_L) \in [q]^L} \left(\prod_{i=1}^{L} P(x_i) \right) \max_{x \in [q]} \sum_{i \in [L]} \omega(i) \mathbb{1}\{x_i = x\}. \end{aligned}$$

We define the shorthand notation

$$\max_\omega(x_1, \cdots, x_L) \coloneqq \max_{x \in [q]} \sum_{\substack{i \in [L] \\ x_i = x}} \omega(i) \tag{16}$$

for any $\omega \in \Delta([L])$ and $(x_1, \ldots, x_L) \in [q]^L$.

The first property that we would like to establish is that $f(P,\omega)$ only increases if ω is replaced by U_L, and furthermore that the maximum is uniquely obtained at U_L if $P(x) > 0$ for all $x \in [q]$. In order to do this, we will regularly "average-out" coordinates of ω and then show that the function value increases (or at least, does not decrease). To be introduce some terminology, for $S \subseteq [L]$ we say that $\overline{\omega}$ is obtained from ω by *averaging-out* the subset S of coordinates if $\overline{\omega}$ is defined as

$$\overline{\omega}(i) = \begin{cases} \frac{\sum_{j\in S}\omega(j)}{|S|} & i \in S \\ \omega(i) & i \notin S \end{cases}.$$

The following lemma gives a simple criterion for establishing that, if $\overline{\omega}$ is obtained from ω by averaging two coordinates, then $f(P,\overline{\omega}) \leqslant f(P,\omega)$, and it furthermore gives a criterion for the inequality to be strict. The main thrust of the proof of Lemma 11 is thus to show that this criterion is always satisfied.

▶ **Lemma 10.** *Let $P \in \Delta([q])$ and $\omega \in \Delta([L])$. Suppose $\omega(L-1) \neq \omega(L)$ and that $\overline{\omega} \in \Delta([L])$ is obtained by averaging-out the last two coordinates of ω. Suppose that for all $(x_1,\ldots,x_L) \in [q]^L$ we have*

$$\frac{1}{2}\left(\max_\omega(x_1,\cdots,x_{L-1},x_L) + \max_\omega(x_1,\cdots,x_L,x_{L-1})\right) \geqslant \max_{\overline{\omega}}(x_1,\cdots,x_{L-1},x_L). \quad (17)$$

Then $f(P,\overline{\omega}) \geqslant f(P,\omega)$.

Furthermore, suppose that additionally there exists $(x_1,\ldots,x_L) \in [q]^L$ with $\prod_{i=1}^L P(x_i) > 0$ such that the inequality in Equation (17) is strict. Then $f(P,\overline{\omega}) > f(P,\omega)$.

Proof. Define $\omega' \in \Delta([L])$ as

$$\omega'(i) = \begin{cases} \omega(i), & i \in [L]\backslash\{L-1,L\} \\ \omega(L), & i = L-1 \\ \omega(L-1), & i = L \end{cases}.$$

That is, ω' is obtained by swapping the last two components of ω. By symmetry, we have $f(P,\omega) = f(P,\omega')$ and so

$$\begin{aligned} f(P,\omega) &= \frac{1}{2}(f(P,\omega) + f(P,\omega')) \\ &= \frac{1}{2}\left(1 - \sum_{(x_1,\cdots,x_L)\in[q]^L}\left(\prod_{i=1}^L P(x_i)\right)\max_\omega(x_1,\ldots,x_{L-1},x_L)\right. \\ &\qquad \left. +1 - \sum_{(x_1,\cdots,x_L)\in[q]^L}\left(\prod_{i=1}^L P(x_i)\right)\max_{\omega'}(x_1,\ldots,x_{L-1},x_L)\right) \\ &= 1 - \sum_{(x_1,\cdots,x_L)\in[q]^L}\left(\prod_{i=1}^L P(x_i)\right)\frac{1}{2}\left(\max_\omega(x_1,\ldots,x_{L-1},x_L) + \max_\omega(x_1,\ldots,x_L,x_{L-1})\right)) \\ &\leqslant 1 - \sum_{(x_1,\cdots,x_L)\in[q]^L}\left(\prod_{i=1}^L P(x_i)\right)\max_{\overline{\omega}}(x_1,\ldots,x_{L-1},x_L) \\ &= f(P,\overline{\omega}), \end{aligned}$$

where the inequality follows from Equation (17). From the above sequence of inequalities, it is also clear that if additionally there exists $(x_1,\ldots,x_L) \in [q]^L$ with $\prod_{i=1}^L P(x_i) > 0$ for which the inequality in Equation (17) is strict, then $f(P,\overline{\omega}) > f(P,\omega)$. ◀

We now establish that the function value cannot decrease if ω is replaced by U_L.

▶ **Lemma 11.** *For any $P \in \Delta([q])$ and $\omega \in \Delta([L])$, $f(P,\omega) \leqslant f(P,U_L)$.*

Proof. Fix any $(x_1,\cdots,x_L) \in [q]^L$. Let $\omega \in \Delta([L])$ be non-uniform. Without loss of generality, assume $\omega(L-1) \neq \omega(L)$. Let $\overline{\omega} \in \Delta([L])$ be obtained by uniformizing the last two components of ω, i.e.,

$$\overline{\omega}(i) = \begin{cases} \omega(i), & i \in [L]\backslash\{L-1,L\} \\ \frac{1}{2}(\omega(L-1)+\omega(L)), & i \in \{L-1,L\} \end{cases}.$$

We claim $f(P,\overline{\omega}) \geqslant f(P,\omega)$. By Lemma 10, we just need to establish Equation (17).

Equation (17) trivially holds if $x_{L-1} = x_L$. We therefore assume below $x_{L-1} \neq x_L$. Let $x_{L-1} = a$ and $x_L = b$. Let

$$\omega^{(a)} = \sum_{\substack{i\in[L-2]\\ x_i=a}} \omega(i), \qquad \omega^{(b)} = \sum_{\substack{i\in[L-2]\\ x_i=b}} \omega(i)\ .$$

Then, we have

$$\sum_{\substack{i\in[L]\\ x_i=a}} \overline{\omega}(i) = \omega^{(a)} + \frac{1}{2}(\omega(L-1)+\omega(L)), \qquad \sum_{\substack{i\in[L]\\ x_i=b}} \overline{\omega}(i) = \omega^{(b)} + \frac{1}{2}(\omega(L-1)+\omega(L)).$$

We first assume that there exists $c \notin \{a,b\}$ such that

$$\max_{\overline{\omega}}(x_1,\cdots,x_L) = \sum_{\substack{i\in[L]\\ x_i=c}} \overline{\omega}(i) = \sum_{\substack{i\in[L]\\ x_i=c}} \omega(i)$$

where the second equality follows since the set $\{i \in [q] : x_i = c\}$ does not contain $L-1, L$. Equation (17) therefore holds as

$$\max_{\omega}(x_1,\cdots,x_{L-1},x_L) \geqslant \sum_{\substack{i\in[L]\\ x_i=c}} \omega(i), \qquad \max_{\omega}(x_1,\cdots,x_L,x_{L-1}) \geqslant \sum_{\substack{i\in[L]\\ x_i=c}} \omega(i).$$

We proceed to the case that

$$\max_{\overline{\omega}}(x_1,\cdots,x_L) = \max\left\{\frac{1}{2}(\omega(L-1)+\omega(L)) + \omega^{(a)}, \frac{1}{2}(\omega(L-1)+\omega(L)) + \omega^{(b)}\right\}.$$

Equation (17) holds as

$$\max_{\omega}(x_1,\cdots,x_{L-1},x_L) \geqslant \max\left\{\omega^{(a)} + \omega(L-1), \omega^{(b)} + \omega(L)\right\}$$

and

$$\max_{\omega}(x_1,\cdots,x_L,x_{L-1}) \geqslant \max\left\{\omega^{(a)} + \omega(L), \omega^{(b)} + \omega(L-1)\right\}.$$

Thus, Lemma 10 implies $f(P,\overline{\omega}) \geqslant f(P,\omega)$, as desired.

We can then continue averaging components of ω and in this way obtain a sequence $(\omega_i)_{i\in\mathbb{N}}$ of distributions with $\omega_1 = \omega$. This sequence converges in ℓ_∞-norm to the uniform distribution U_L and satisfies $f(P,\omega_{i+1}) \geqslant f(P,\omega_i)$ for all $i \in \mathbb{N}$. Observing that $\omega \mapsto f(P,\omega)$ is a continuous function – the term $\max_{x\in[q]} \sum_{i\in[L]} \omega(i)\mathbb{1}\{x_i = x\}$ is a maximum over linear functions of ω, hence linear, implying that $f(P,\cdot)$ is a linear combination of continuous functions – it follows that $f(P,U_L) = \lim_{i\to\infty} f(P,\omega_i)$, and in particular that $f(P,U_L) \geqslant f(P,\omega_1) = f(P,\omega)$, as desired. ◀

We now strengthen the conclusion of Lemma 11 by showing that for all $q \geqslant 3$ and $L \geqslant 2$ the function $\omega \mapsto f(P,\omega)$ is *uniquely* maximized by the setting $\omega = U_L$, except for degenerate cases where $P(x) = 0$ for some $x \in [q]$.

Before stating and proving this fact, we note that the proof of Lemma 11 in fact shows that we can average out any subset of coordinates of ω and only increase the value of $f(P,\omega)$. We formalize this fact in the following lemma, which will be useful in the following arguments.

▶ **Lemma 12.** *Let $P \in \Delta([q])$, $\omega \in \Delta([L])$ and $S \subseteq [L]$. Let $\overline{\omega}$ be obtained from ω by averaging-out the subset S of coordinates. Then $f(P,\overline{\omega}) \geqslant f(P,\omega)$.*

▶ **Theorem 13.** *Let $q \geqslant 3$, $L \geqslant 2$ and let $P \in \Delta([q])$ be such that $P(x) > 0$ [6] for all $x \in [q]$. Then for all $\omega \in \Delta([L])$, $f(P,\omega) \leqslant f(P,U_L)$ with equality if and only if $\omega = U_L$.*

Proof. The inequality was already established in Lemma 11, so we focus on showing $\omega = U_L$ when $f(P,\omega) = f(P,U_L)$. As $q \geqslant 3$, let a, b and c denote 3 distinct elements of $[q]$. Let $\omega \neq U_L$ and suppose for a contradiction that $f(P,\omega)$ is a maximum of the function $\omega \mapsto f(P,\omega)$. The proof proceeds via a number of cases.

1. ***L* is even.** Without loss of generality, $\omega(L-1) < \omega(L)$. If $L \geqslant 4$, let ω' be obtained from ω by averaging-out the first $L-2$ coordinates; by Lemma 12, $f(P,\omega') \geqslant f(P,\omega)$. If $L = 2$, set $\omega' = \omega$.
   If $L \geqslant 4$, since $2|(L-2)$, we can set $x_1 = \cdots = x_{L/2-1} = a$ and $x_{L/2} = \cdots = x_{L-2} = b$. Set further $x_{L-1} = a$ and $x_L = b$. We observe that for this $(x_1,\ldots,x_L) \in [q]^L$ and $\overline{\omega}$ obtained from ω' by averaging-out the last two coordinates, Equation (17) strictly holds. Indeed,

$$\max_{\omega'}(x_1,\ldots,x_{L-1},x_L) + \max_{\omega'}(x_1,\ldots,x_L,x_{L-1}) = 2\sum_{j=1}^{(L-2)/2}\omega(i) + 2\omega(L)$$

   and

$$2\max_{\overline{\omega}}(x_1,\ldots,x_{L-1},x_L) = 2\sum_{j=1}^{(L-2)/2}\omega(i) + \omega(L-1) + \omega(L).$$

   Thus Lemma 10 implies $f(P,\overline{\omega}) > f(P,\omega') \geqslant f(P,\omega)$, a contradiction.
2. ***L* is odd and at least three components of *ω* take distinct values, or the components in *ω* only take two different values and at least two of them take the minimum value.** Without loss of generality $\omega(L-2) \leqslant \omega(L-1) < \omega(L)$. If $L \geqslant 5$, let ω' be obtained from ω by averaging-out the first $L-3$ coordinates; by Lemma 12, $f(P,\omega') \geqslant f(P,\omega)$. If $L = 3$, set $\omega' = \omega$.
   Since $2|(L-3)$, if $L \geqslant 5$, we set $x_1 = \cdots = x_{(L-1)/2-1} = a$ and $x_{(L-1)/2} = \cdots = x_{L-3} = b$. Let $x_{L-2} = c$, $x_{L-1} = a$ and $x_L = b$. We observe that for this $(x_1,\ldots,x_L) \in [q]^L$ and $\overline{\omega}$ obtained from ω' by averaging-out the last two coordinates, Equation (17) strictly holds. Indeed, since $\omega(L-2) < \omega(L)$ we have

$$\max_{\omega'}(x_1,\ldots,x_{L-1},x_L) + \max_{\omega'}(x_1,\ldots,x_L,x_{L-1}) = 2\sum_{j=1}^{(L-2)/2}\omega(i) + 2\omega(L)$$

[6] In fact, our proof only apply with $P = U_q$ which clearly satisfies the condition.

and

$$2\max_{\overline{\omega}}(x_1,\ldots,x_{L-1},x_L) = 2\sum_{j=1}^{(L-3)/2}\omega(i)+\omega(L-1)+\omega(L).$$

Thus Lemma 10 implies $f(P,\overline{\omega}) > f(P,\omega') \geqslant f(P,\omega)$, a contradiction.

3. ***L* is odd and only one component takes the minimum value.** That is, $\omega(1) = \omega(2) = \cdots = \omega(L-1) < \omega(L)$. Let ω' be obtained from ω' by averaging-out the subset $\{L-1, L_2\}$. Then $f(P,\omega') \geqslant f(P,\omega)$ by Lemma 12 and moreover ω' is such that at least two coordinates take on the minimum value, as $\omega'(1) = \cdots = \omega'(L-2) > \omega'(L-1) = \omega'(L)$. The argument from the previous case can now be applied to derive a contradiction. ◀

Thus, except for degenerate choices for $P \in \Delta([q])$, it follows that the function $\omega \mapsto f(P,\omega)$ is maximized by the choice of $\omega = U_L$. The next step is to determine the distribution $P \in \Delta([q])$ maximizing $P \mapsto f(P,U_L)$. At this point, we can rely on a main result of [25], as the function $P \mapsto 1 - f(P,U_L)$ is the same as the function $f_{q,L}(P)$ defined in [25, Equation (17)]. It is shown therein that $f_{q,L}(P)$ is *strictly Schur convex*, which in particular means that $f_{q,L}(P)$ has a unique minimum at $P = U_q$. That is, $f(P,U_L)$ has a unique maximum at $P = U_q$.

The (strict) Schur convexity also implies the following: if $p = \max_{x\in[q]} P(x)$, then $f_{q,L}(P) \geqslant f_{q,L}(P_{q,p})$ where

$$P_{q,p}(x) = \begin{cases} \frac{1-p}{q-1} & x \in \{1,2,\ldots,q-1\} \\ p & x = q \end{cases} . \tag{18}$$

That is, we can conclude that $f(P,U_L) \leqslant f(P_{q,p},U_L)$. We encapsulate these facts in the following proposition.

▶ **Proposition 14** (Theorem 1,2 [25]). *Let $q \geqslant 2$, $L \geqslant q$ and $P \in \Delta([q])$. Suppose $p = \max_{x\in[q]} P(x)$. Then $f(P,U_L) \leqslant f(P_{q,p},U_L)$. Furthermore, $f(P_{q,p},U_L) \leqslant f(U_q,U_L)$ is monotone decreasing for $p \geqslant 1/q$. Lastly, $f(P_{q,p},U_L)$ is concave for $p \in [1/q,1]$, i.e., $\frac{1}{n}\sum_{i=1}^n f(P_{q,p_i},U_L) \leqslant f(P_{q,p},U_L)$ with $p = \frac{1}{n}\sum_{i=1}^n p_i$.*

A further fact that we have from [25] is that

$$p_*(q,L) = f(U_q,U_L) .$$

In fact, this was taken as the definition of $p_*(q,L)$. To end this subsection, we prove the following theorem by utilizing the concavity of $f(P_{q,p},U_L)$.

▶ **Theorem 15.** *Assume $\mathrm{rad}_{\mathrm{H}}(\mathcal{C}) \leqslant p$, then we have*

$$\mathop{\mathbb{E}}_{(\boldsymbol{c}_1,\ldots,\boldsymbol{c}_L)\in\mathcal{C}^L}[\mathrm{rad}_\omega(\varphi(\boldsymbol{c}_1),\ldots,\varphi(\boldsymbol{c}_L))] \leqslant f(P_{q,p},U_L).$$

Proof. Let $\boldsymbol{y}$ be the center attaining $\mathrm{rad}_{\mathrm{H}}(\mathcal{C})$. Without loss of generality, we can assume $\boldsymbol{y}$ is a all zero vector. Let P_i be the distribution of symbols in the i-th index of $\mathcal{C}$, i.e., $P_i(j) = \Pr[\boldsymbol{c}(i) = j]$ with the distribution taken over $\boldsymbol{c} \in \mathcal{C}$. Let $p_i = \max_{x\in[q]} P_i(x)$ and $p' = \frac{1}{n}\sum_{i=1}^n p_i$. Clearly, $p_i \geqslant 1/q$. Then, we have

$$\begin{aligned}\mathop{\mathbb{E}}_{(\boldsymbol{c}_1,\ldots,\boldsymbol{c}_L)\in\mathcal{C}^L}[\mathrm{rad}_\omega(\varphi(\boldsymbol{c}_1),\ldots,\varphi(\boldsymbol{c}_L))] &= \frac{1}{n}\sum_{i=1}^n f(P_i,\omega) \\ \leqslant \frac{1}{n}\sum_{i=1}^n f(P_i,U_L) &\leqslant \frac{1}{n}\sum_{i=1}^n f(P_{q,p_i},U_L) \leqslant f(P_{q,p'},U_L) \leqslant f(P_{q,p},U_L).\end{aligned}$$

The first inequality is due to Lemma 11 and the second and third inequalities are due to Proposition 14. The last inequality is due to $\mathrm{rad}_{\mathrm{H}}(\mathcal{C}) \leqslant p$ and the center $\boldsymbol{y}$ is all zero vector. The proof is completed. ◀

3.3 Abundance of Random-Like L-tuples

Recall the notations

$$\mathrm{rad}(\mathcal{C}) = \frac{1}{n} \min_{\boldsymbol{y}\in[q]^n} \max_{\boldsymbol{c}\in\mathcal{C}} d_{\mathrm{H}}(\boldsymbol{c}, \boldsymbol{y})$$

and

$$\mathsf{type}_{\boldsymbol{u}}(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_L) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{(\boldsymbol{c}_1(i), \ldots, \boldsymbol{c}_L(i)) = \boldsymbol{u}\}$$

where $\boldsymbol{c}_i = (\boldsymbol{c}_i(1), \ldots, \boldsymbol{c}_i(n)) \in [q]^n$ and $\boldsymbol{u} \in [q]^L$. In this subsection, we prove a code $\mathcal{C} \subseteq [q]^n$ either contains a large subcode $\mathcal{C}' \subseteq \mathcal{C}$ with radius $\mathrm{rad}(\mathcal{C}') \leqslant 1 - \frac{1}{q} - \varepsilon$, or most of its L-tuples are of type close to the uniform distribution (for all $\boldsymbol{u} \in [q]^L$).

We first show that for any projection π_A with $|A| \geqslant \mu n$ (for some parameter $\mu \in [0, 1]$), the projection $\pi_A(\mathcal{C})$ almost preserves the radius $\mathrm{rad}(\mathcal{C})$ with small loss. Then, if $\mathrm{rad}(\mathcal{C}') > 1 - \frac{1}{q} - \varepsilon$ for any subcode $\mathcal{C}'$ with large size, we find a codeword $\boldsymbol{c}_1$ in $\mathcal{C}$ whose symbols' distribution is close to the uniform. In fact, most codewords in $\mathcal{C}$ satisfies this requirement. Let A_i be the index set of $\boldsymbol{c}_1$ taking value i. We apply π_{A_i} to $\mathcal{C}$ and claim that $\pi_{A_i}(\mathcal{C})$ preserves the radius. Thus, we can find a codeword $\boldsymbol{c}_2$ such that for every $i \in [q]$, the symbol's distribution of $\pi_{A_i}(\boldsymbol{c}_2)$ is close to uniform. Moreover, most of codewords in $\mathcal{C}$ satisfy this requirement. The proof is the completed by induction.

▶ **Lemma 16.** *Let $\pi_A : [q]^n \to [q]^A$ be the projection on a set A of size m. Suppose $\mathcal{C} \subseteq [q]^n$ is a code of size qs satisfying $\mathrm{rad}_{\mathrm{H}}(\pi_A(\mathcal{C})) \leqslant 1 - \frac{1}{q} - \varepsilon$. Then, there exists a subcode $\mathcal{C}' \subseteq \mathcal{C}$ of size at least s such that $\mathrm{rad}_{\mathrm{H}}(\mathcal{C}') \leqslant 1 - \frac{1}{q} - \frac{m}{n}\varepsilon$.*

Proof. The proof is omitted due to the limited space. ◀

▶ **Theorem 17.** *Let L be fixed. For every $\varepsilon > 0$, there exists a $\delta > 0$ with the following property. If s is a natural number, there exist constants $M_0 = M_0(s)$ and $c(s)$ such that for any code $\mathcal{C} \subseteq [q]^n$ with size $M \geqslant M_0$, at least one of the following must hold:*

1. *There exists $\mathcal{C}' \subseteq \mathcal{C}$ such that $|\mathcal{C}'| \geqslant s$ and $\mathrm{rad}_{\mathrm{H}}(\mathcal{C}') \leqslant 1 - \frac{1}{q} - \delta$.*
2. *There exist at least $M^L - c(s)M^{L-1}$ many L-tuples of distinct codewords $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_L)$ in $\mathcal{C}$ such that for all $\boldsymbol{u} \in [q]^L$ we have*

$$|\mathsf{type}_{\boldsymbol{u}}(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_L) - q^{-L}| \leqslant \varepsilon$$

and thus

$$|\overline{\mathrm{rad}}_{\omega}(\varphi(\boldsymbol{c}_1), \cdots, \varphi(\boldsymbol{c}_L)) - f(U_q, \omega)| \leqslant q^L\varepsilon.$$

Proof. The proof is omitted due to the limited space. ◀

3.4 Putting Everything Together

The argument follows the same line of reasoning as [1]. We provide the proof for completeness. Define $\rho_L(\mathcal{C}) = \min \mathrm{rad}(\varphi(\boldsymbol{c}_1), \ldots, \varphi(\boldsymbol{c}_L))$ with minimum taken over all L-tuples $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_L) \in \mathcal{C}^L$ with distinct elements, where we recall that rad denotes the *relaxed* Chebyshev radius (Definition 5).

▶ **Theorem 18.** *Let $L \geqslant 2$ and $q \geqslant 3$. If $\mathcal{C} \subseteq [q]^n$ is $(p_*(q, L) + \varepsilon, L)$-list-decodable, then $|\mathcal{C}| = O_{q,L}(\frac{1}{\varepsilon})$.*

Proof. The proof is omitted due to the limited space. ◀

4 Zero-Rate List-Recovery

In this section, we show how our results on list-decoding can naturally be extended to list-recovery. Due to the page limit, we refer the reader to the full version. The result is summarized in Theorem 1.

5 Code Construction

In this section, we present a simple simplex-like code construction and show that it attains the optimal size-radius trade-off by analyzing its list-decoding and -recovery radius.

Our construction will be identical for list-decoding and -recovery and therefore we will directly analyze its list-recovery radius. Before presenting the construction and its analysis, let us define the average radius $\overline{\mathrm{rad}}_\ell$. This is a standard notion that "linearizes" the Chebyshev radius rad and often finds its use in the analysis of list-recoverable codes in the literature. The definition reads as follows: for any $\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L \in [q]^n$,

$$\overline{\mathrm{rad}}_\ell(\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L) \coloneqq \frac{1}{L} \min_{\boldsymbol{Y} \in \mathcal{X}^n} \sum_{i=1}^{L} d_{\mathrm{LR}}(\boldsymbol{c}_i, \boldsymbol{Y}).$$

It is well-known and easy to verify (by, e.g., following the derivations leading to Equation (14)) that the above minimization admits the following explicit solution:

$$\overline{\mathrm{rad}}_\ell(\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L) \coloneqq \sum_{j=1}^{n} \left(1 - \frac{1}{L} \mathsf{pl}_\ell(\boldsymbol{c}_1(j), \cdots, \boldsymbol{c}_L(j))\right), \tag{19}$$

Equation (19) should be interpreted as the average distance from each $\boldsymbol{c}_i$ to the "centroid" $\boldsymbol{Y}^* \in \mathcal{X}^n$ of the list defined as[7]

$$\boldsymbol{Y}^*(j) \coloneqq \operatorname*{argmax}_{A \in \mathcal{X}} \sum_{i=1}^{L} \mathbb{1}\{\boldsymbol{c}_i(j) \in A\}.$$

for each $j \in [n]$.

Finally, for integers $q \geqslant 1$ and $L \geqslant 0$, denote by

$$\mathcal{A}_{q,L} = \left\{(a_1, \cdots, a_q) \in \mathbb{Z}_{\geqslant 0}^q : \sum_{i=1}^{q} a_i = L\right\}$$

[7] If there are multiple maximizers, take an arbitrary one and the value of $\overline{\mathrm{rad}}_\ell$ remains the same.

the set of q-partitions of L, i.e., a_i is the number of indices taking value i. For $\boldsymbol{a} \in \mathcal{A}_{a,L}$, we shorthand $\binom{L}{\boldsymbol{a}} = \binom{L}{a_1,\ldots,a_q}$ where $\boldsymbol{a} = (a_1, \ldots, a_q)$. Define $\max_\ell\{\boldsymbol{a}\} = \max_{A\in\mathcal{X}} \sum_{i\in A} a_i$, i.e., the sum of ℓ largest components in $\boldsymbol{a}$.

▶ **Theorem 19** (Construction of zero-rate list-recoverable codes). *Fix any integers $q \geqslant 3$, $\ell \geqslant 1$ and $L \geqslant 2$. For any sufficiently large m, there exists a (p, ℓ, L)-list-recoverable code $\mathcal{C}$ with blocklength*

$$n = \binom{qm}{\underbrace{m, \cdots, m}_{q}}, \tag{20}$$

and the trade-off between code size M and (relative) radius p given by:

$$M = qm, \quad p = p_*(q, \ell, L) + c_{q,\ell,L} m^{-1} + O(m^{-2}),$$

where

$$c_{q,\ell,L} \coloneqq q^{-L} \sum_{\boldsymbol{a}\in\mathcal{A}_{q,L}} \frac{\max_\ell\{\boldsymbol{a}\}}{L} \binom{L}{\boldsymbol{a}} \left[\sum_{i=1}^{q} \binom{a_i}{2} - \frac{1}{q}\binom{L}{2}\right] > 0. \tag{21}$$

Proof. The proof is omitted due to the page limit. ◀

References

1 Noga Alon, Boris Bukh, and Yury Polyanskiy. List-decodable zero-rate codes. *IEEE Transactions on Information Theory*, 65(3):1657–1667, 2018. doi:10.1109/TIT.2018.2868957.

2 László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Comput. Complex.*, 3:307–318, 1993. doi:10.1007/BF01275486.

3 L. A. Bassalygo. New upper bounds for error-correcting codes. *Probl. of Info. Transm.*, 1:32–35, 1965.

4 Vladimir M Blinovsky. Bounds for codes in the case of list decoding of finite volume. *Problems of Information Transmission*, 22:7–19, 1986.

5 Vladimir M Blinovsky. Code bounds for multiple packings over a nonbinary finite alphabet. *Problems of Information Transmission*, 41:23–32, 2005. doi:10.1007/S11122-005-0007-5.

6 Vladimir M Blinovsky. On the convexity of one coding-theory function. *Problems of Information Transmission*, 44:34–39, 2008. doi:10.1134/S0032946008010031.

7 Philippe Delsarte. An algebraic approach to the association schemes of coding theory. *Philips Res. Rep. Suppl.*, 10:vi+–97, 1973.

8 Dean Doron, Dana Moshkovitz, Justin Oh, and David Zuckerman. Nearly optimal pseudorandomness from hardness. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1057–1068. IEEE, 2020. doi:10.1109/FOCS46700.2020.00102.

9 Dean Doron and Mary Wootters. High-probability list-recovery, and applications to heavy hitters. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ICALP.2022.55.

10 Peter Elias. List decoding for noisy channels. *Wescon Convention Record, Part 2*, pages 94–104, 1957.

11 Peter Elias. Error-correcting codes for list decoding. *IEEE Transactions on Information Theory*, 37(1):5–12, 1991. doi:10.1109/18.61123.

12 Edgar N Gilbert. A comparison of signalling alphabets. *The Bell System Technical Journal*, 31(3):504–522, 1952.

13 Oded Goldreich and Leonid A Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 25–32. ACM, 1989. doi:10.1145/73007.73010.

14 Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from parvaresh–vardy codes. *Journal of the ACM (JACM)*, 56(4):1–34, 2009. doi:10.1145/1538902.1538904.

15 Iftach Haitner, Yuval Ishai, Eran Omri, and Ronen Shaltiel. Parallel hashing via list recoverability. In *Annual Cryptology Conference*, pages 173–190. Springer, 2015. doi:10.1007/978-3-662-48000-7_9.

16 Justin Holmgren, Alex Lombardi, and Ron D Rothblum. Fiat–shamir via list-recoverable codes (or: parallel repetition of gmw is not zero-knowledge). In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 750–760, 2021. doi:10.1145/3406325.3451116.

17 Piotr Indyk, Hung Q Ngo, and Atri Rudra. Efficiently decodable non-adaptive group testing. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1126–1142. SIAM, 2010. doi:10.1137/1.9781611973075.91.

18 Jeffrey C Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55(3):414–440, 1997. doi:10.1006/JCSS.1997.1533.

19 Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the Fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993. doi:10.1137/0222080.

20 VI Levenshtein. Application of hadamard matrices on coding problem. *Problems of Cybernetica*, 5:123–136, 1961.

21 Richard J Lipton. Efficient checking of computations. In *Proceedings of the 7th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 207–215. Springer, 1990. doi:10.1007/3-540-52282-4_44.

22 Robert J. McEliece, Eugene R. Rodemich, Howard Rumsey, Jr., and Lloyd R. Welch. New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities. *IEEE Trans. Inform. Theory*, IT-23(2):157–166, 1977. doi:10.1109/tit.1977.1055688.

23 Hung Q Ngo, Ely Porat, and Atri Rudra. Efficiently decodable error-correcting list disjunct matrices and applications. In *International Colloquium on Automata, Languages, and Programming*, pages 557–568. Springer, 2011.

24 Nicolas Resch. List-decodable codes:(randomized) constructions and applications. *School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep., CMU-CS-20-113*, 2020.

25 Nicolas Resch, Chen Yuan, and Yihan Zhang. Zero-rate thresholds and new capacity bounds for list-decoding and list-recovery. *arXiv preprint*, 2022. doi:10.48550/arXiv.2210.07754.

26 Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001. doi:10.1006/JCSS.2000.1730.

27 Michael A Tsfasman, SG Vlădutx, and Th Zink. Modular curves, shimura curves, and goppa codes, better than varshamov-gilbert bound. *Mathematische Nachrichten*, 109(1):21–28, 1982.

28 RR Varshamov. Estimate of the number of signals in error correcting codes. *Docklady Akad. Nauk, SSSR*, 117:739–741, 1957.

29 Lloyd R. Welch, Robert J. McEliece, and Howard Rumsey, Jr. A low-rate improvement on the Elias bound. *IEEE Trans. Inform. Theory*, IT-20:676–678, 1974. doi:10.1109/tit.1974.1055279.

30 Jack Wozencraft. List decoding. *Quarter Progress Report*, 48:90–95, 1958.

Fine-Grained Equivalence for Problems Related to Integer Linear Programming

Lars Rohwedder
University of Southern Denmark (SDU), Odense, Denmark

Karol Węgrzycki
Saarland University, Saarbrücken, Germany
Max Planck Institute for Informatics, Saarbrücken, Germany

Abstract

Integer Linear Programming with n binary variables and m many 0/1-constraints can be solved in time $2^{\tilde{O}(m^2)}\text{poly}(n)$ and it is open whether the dependence on m is optimal. Several seemingly unrelated problems, which include variants of Closest String, Discrepancy Minimization, Set Cover, and Set Packing, can be modelled as Integer Linear Programming with 0/1 constraints to obtain algorithms with the same running time for a natural parameter m in each of the problems. Our main result establishes through fine-grained reductions that these problems are equivalent, meaning that a $2^{O(m^{2-\varepsilon})}\text{poly}(n)$ algorithm with $\varepsilon > 0$ for one of them implies such an algorithm for all of them.

In the setting above, one can alternatively obtain an $n^{O(m)}$ time algorithm for Integer Linear Programming using a straightforward dynamic programming approach, which can be more efficient if n is relatively small (e.g., subexponential in m). We show that this can be improved to ${n'}^{O(m)} + O(nm)$, where n' is the number of distinct (i.e., non-symmetric) variables. This dominates both of the aforementioned running times.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Integer Programming, Fine-Grained Complexity, Fixed-Parameter Tractable Algorithms

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.83

Related Version *Full Version*: https://arxiv.org/abs/2409.03675

Funding *Lars Rohwedder*: Supported by Dutch Research Council (NWO) project "The Twilight Zone of Efficiency: Optimality of Quasi-Polynomial Time Algorithms" [grant number OCEN.W.21.268]. *Karol Węgrzycki*: This work is part of the project TIPEA that has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No. 850979).

Acknowledgements We thank Friedrich Eisenbrand for his valuable insights and constructive discussions that enhanced this work.

1 Introduction

The study of parameterized complexity for Integer Linear Programming has a long history: classical works by Lenstra [15] and Kannan [12] and very recently Rothvoss and Reis [19] provide FPT algorithms in the number of variables n of an ILP of the form $Ax \leq b,\ x \in \mathbb{Z}^n$. In an orthogonal line of research, Papadimitriou [17] gave an FPT algorithm in the number of constraints and the size of the coefficients of A and b for an ILP in the standard form $Ax = b,\ x \in \mathbb{Z}^n_{\geq 0}$. Interest in the second line of work has been renewed by the improved algorithms due to Eisenbrand, Weismantel [9] and Jansen, Rohwedder [11], which give essentially optimal running times $(m\Delta)^{O(m)}\,\text{poly}(n)$ due to a conditional lower bound based on the Exponential Time Hypothesis (ETH) [14]. Here, Δ is the maximum absolute size

© Lars Rohwedder and Karol Węgrzycki;
licensed under Creative Commons License CC-BY 4.0

of an entry in A. The work by Eisenbrand and Weismantel also considers a version where variables are subject to box-constraints, which will be the primary focus of this work, see the definition below.

Integer Linear Programming
Input: Constraint matrix $A \in \{-\Delta, \dots, \Delta\}^{m \times n}$, right-hand side $b \in \mathbb{Z}^m$, variable bounds $\ell, u \in \mathbb{Z}^n_{\geq 0}$.
Task: Find $x \in \mathbb{Z}^n$ with

$$Ax = b$$
$$\ell_i \leq x_i \leq u_i \qquad i = 1, 2, \dots, n.$$

We refer to the variant where $\ell = (0, \dots, 0)^\mathsf{T}$ and $u = (1, \dots, 1)^\mathsf{T}$ as *binary* Integer Linear Programming.

Binary Integer Linear Programming
Input: Constraint matrix $A \in \{-\Delta, \dots, \Delta\}^{m \times n}$, right-hand side $b \in \mathbb{Z}^m$.
Task: Find $x \in \{0, 1\}^n$ with

$$Ax = b.$$

The running times obtained in [9] for either variant is $(m\Delta)^{O(m^2)} \operatorname{poly}(n)$. Also for matrices with only 0/1 coefficients nothing better than $2^{\tilde{O}(m^2)} \operatorname{poly}(n)$ is known. It is an intriguing question whether the slightly unusual exponent of $\tilde{O}(m^2)$ is necessary, which is in the spirit of *fine-grained complexity*. Since the dominant complexity-theoretic assumption of P≠NP is not powerful enough to show precise lower bounds on running times, the field of fine-grained complexity is concerned with finding lower bounds via stronger conjectures, see e.g., [2, 20, 6]. A number of such conjectures exist by now, often with interesting connections between them. Even if one doubts these conjectures, the reductions still provide insights into how problems relate to each other.

Based on existing conjectures, the best lower bound known on the exponent of Integer Linear Programming is $\Omega(m \log m)$ from the easier unbounded setting [14], which is of course not tight in this setting. In this paper, we take another path: we assume that Integer Linear Programming cannot be solved faster than the state-of-the-art and present several other natural parameterized problems that are all equivalent with respect to improvements on their running time.

▶ **Hypothesis 1** (ILP Hypothesis). *For every fixed $\varepsilon > 0$, there is no $2^{O(m^{2-\varepsilon})} \operatorname{poly}(n)$ time algorithm for Integer Linear Programming with $\Delta = O(1)$.*

In all of the problems below, the symbol m is chosen for the parameter of interest.

Closest String with Binary Alphabet
Input: Alphabet $\Sigma = \{0, 1\}$, strings $s_1, s_2, \dots, s_m \in \Sigma^n$
Task: Find string $t \in \Sigma^n$ minimizing

$$\max_i d(t, s_i),$$

where $d(t, s_i)$ is the Hamming distance between t and s_i, i.e., the number of positions the two strings differ in.

We refer to the generalization with arbitrary Σ simply as Closest String.

Discrepancy Minimization
Input: Universe $U = \{1, 2, \ldots, n\}$, set system $S_1, S_2, \ldots, S_m \subseteq U$
Task: Find coloring $\chi : U \to \{-1, 1\}$ minimizing

$$\max_i \left| \sum_{u \in S_i} \chi(u) \right|.$$

Set Multi-Cover
Input: Universe $U = \{1, 2, \ldots, m\}$, set system $S_1, S_2, \ldots, S_n \subseteq U$, $b \in \mathbb{N}$
Task: Find $I \subseteq \{1, 2, \ldots, n\}$ of minimal cardinality such that for each $v \in U$ there are at least b sets S_i with $i \in I$ and $v \in S_i$.

Set Multi-Packing
Input: Universe $U = \{1, 2, \ldots, m\}$, set system $S_1, S_2, \ldots, S_n \subseteq U$, $b \in \mathbb{N}$
Task: Find $I \subseteq \{1, 2, \ldots, n\}$ of maximal cardinality such that for each $v \in U$ there are at most b sets S_i with $i \in I$ and $v \in S_i$.

Many of these problems are well-known applications of ILP techniques. For example, Knop et al. [13] provided $m^{O(m^2)} \cdot \text{poly}(n)$ algorithms for Closest String and Set Multi-Cover. Discrepancy Minimization is usually considered within the context of approximation algorithms (see, e.g., [3]).

As mentioned above, our main result is the following equivalence.

▶ **Theorem 2.** *The following statements are equivalent:*
(1) *There exists an* $2^{O(m^{2-\varepsilon})} \text{poly}(n)$ *algorithm for Integer Linear Programming with* $\Delta = O(1)$ *with* $\varepsilon > 0$.
(2) *There exists an* $2^{O(m^{2-\varepsilon})} \text{poly}(n)$ *algorithm for Binary Integer Linear Programming with* $A \in \{0, 1\}^{m \times n}$ *and* $n \leq m^{O(m)}$ *with* $\varepsilon > 0$.
(3) *There exists an* $2^{O(m^{2-\varepsilon})} \text{poly}(n)$ *algorithm for Closest String with Binary Alphabet with* $\varepsilon > 0$.
(4) *There exists an* $2^{O(m^{2-\varepsilon})} \text{poly}(n)$ *algorithm for Discrepancy Minimization with* $\varepsilon > 0$.
(5) *There exists an* $2^{O(m^{2-\varepsilon})} \text{poly}(n)$ *algorithm for Set Multi-Cover with* $\varepsilon > 0$.
(6) *There exists an* $2^{O(m^{2-\varepsilon})} \text{poly}(n)$ *algorithm for Set Multi-Packing with* $\varepsilon > 0$.

Note that Item 1 is the negation of Hypothesis 1. All problems in Theorem 2 are easily transformed into the first problem, i.e., Integer Linear Programming with $\Delta = O(1)$, while maintaining the same value of m. Hence, the more interesting aspect of the theorem is that all these problems are as expressive as the first one.

Hypothesis 1 considers Integer Linear Programming with relatively small entries, i.e., $\Delta = O(1)$. One can also ask the question of whether there is any parameter regime for Δ for which the state-of-the-art can be improved. In this spirit, a stronger variant of the conjecture is the following.

▶ **Hypothesis 3** (Strong ILP Hypothesis). *For every fixed* $\varepsilon > 0$ *and* $\delta \geq 0$, *there is no* $2^{O(m^{\delta+2-\varepsilon})} \text{poly}(n)$ *time algorithm for Integer Linear Programming with* $\Delta = 2^{m^\delta}$.

Note that Hypothesis 1 is a special case of Hypothesis 3 for $\delta = 0$. Another interesting regime is the complexity of Integer Linear Programming with $\Delta = 2^m$, because of a connection to block-structure integer programming, which we elaborate on later. There, the state-of-the-art algorithm requires time $m^{O(m^3)} \text{poly}(n)$, Integer Linear Programming with large entries can be reduced to an equivalent instance with a 0/1 matrix as seen in the following theorem, but the reduction is not strong enough to show equivalence between the two hypotheses.

▶ **Theorem 4.** *There is a polynomial time algorithm that transforms an instance of* Integer Linear Programming *with* $\Delta > 1$ *into an equivalent one with* $A' \in \{0,1\}^{m' \times n'}$ *for* $m' = O(m \log \Delta)$ *and* $n' \leq {m'}^{O(m')}$.

This implies that if there is an algorithm with running time $2^{O(m^{1.5-\varepsilon})} \operatorname{poly}(n)$ for Integer Linear Programming with $A \in \{0,1\}^{m \times n}$, then there is a $2^{O(m^{3-\varepsilon'})} \operatorname{poly}(n)$ time algorithm for Integer Linear Programming with $\Delta = 2^m$.

One might hope to improve the theorem to $m' = O(m\sqrt{\log \Delta})$, since then a $2^{O(m^{2-\varepsilon})} \operatorname{poly}(n)$ time algorithm for 0/1 matrices would imply a $2^{O(m^{3-\varepsilon'})} \operatorname{poly}(n)$ time algorithm for $\Delta = 2^m$. However, such a reduction would imply the strong result that under ETH is equivalent to Hypothesis 1. This is because under ETH the Subset Sum problem, i.e., the case when $m = 1$, cannot be solved in $\Delta^{o(1)} \operatorname{poly}(n)$ time [1] and the hypothetical reduction would be able to encode an instance of Subset Sum into an ILP with $m = O(\sqrt{\log \Delta})$. We are not aware of any meaningful reduction in the other direction, i.e., from large m and small Δ to smaller m and larger Δ. It is possible to aggregate m constraints into a single one with entries bounded by $\Delta' = \Delta^{O(m^2)}$, but this reduction seems useless since the resulting parameter range requires $\operatorname{poly}(\Delta') \cdot \operatorname{poly}(n)$ time due to the ETH lower bound mentioned above.

Assuming Hypothesis 3, we derive a tight lower for a form of block-structured integer linear programs that has been studied extensively in recent literature. For simplicity, we consider here the basic setting with $m \times m$ submatrices.

n-Fold Integer Linear Programming
Input: Square matrices $A_1, \ldots, A_n, B_1, \ldots, B_n \in \{-\Delta, \ldots, \Delta\}^{m \times m}$, right-hand sides $b^{(0)}, \ldots, b^{(n)} \in \mathbb{Z}^m$.
Task: Find $x^{(1)}, \ldots, x^{(n)} \in \mathbb{Z}^m_{\geq 0}$ with

$$\begin{aligned} A_1 x^{(1)} + \ldots + A_n x^{(n)} &= b^{(0)} \\ B_1 x^{(1)} &= b^{(1)} \\ &\vdots \\ B_n x^{(n)} &= b^{(n)} \end{aligned}$$

▶ **Theorem 5.** *For every* $\delta > 0$*, there is no algorithm with running time* $2^{O(m^{3-\delta})} \operatorname{poly}(n)$ *for* n*-Fold Integer Linear Programming when the maximum absolute entry is bounded by* $\Delta = O(1)$*, unless Hypothesis 3 is false.*

This matches the best algorithms known for the problem, see [5] and references therein. The reduction follows the same idea as used in [10], where the authors show a non-tight quadratic lower bound for the exponent based on ETH. Our lower bound is stronger simply because the conjecture we base it on is stronger.

1.1 Tightness of more general problems

There are several other, more general problems to the ones mentioned above, for which known algorithms would be tight assuming that one cannot improve the running time for the class of problems in Theorem 2. However, we do not know if these are all equivalent.

The algorithm by Eisenbrand and Weismantel [9] also works for the optimization version of ILP, i.e.,

$$\max c^\mathsf{T} x,\ Ax = b,\ \ell \le x \le u,\ x \in \mathbb{Z}^n.$$

This leads to the same running time of $(m\Delta)^{O(m^2)}\operatorname{poly}(n)$ except for a slightly higher constant in the exponent. Notably, the coefficients of c do not increase the number of arithmetic operations.

Given a matrix $A \in \{-\Delta, \dots, \Delta\}^{m\times n}$ and a convex function $g : \mathbb{R}^m \to \mathbb{R} \cup \{\infty\}$, Dadush, Léonard, Rohwedder, and Verschae [7] have shown that one can find the minimizer of $g(Ax),\ x \in \{0,1\}^n$ in time $(m\Delta)^{O(m^2)}\operatorname{poly}(n)$ (assuming polynomial time to evaluate g). This problem is a generalization of Binary Integer Linear Programming, since one can take $g(b') = 0$ if $b' = b$ and $g(b') = \infty$ otherwise.

Given a linear matroid $M = (E, \mathcal{I})$, where $|E| = n$, a matrix $A \in \{-\Delta, \dots, \Delta\}^{n\times m}$ and a right-hand side $b \in \mathbb{Z}^m$, Eisenbrand, Rohwedder, and Węgrzycki [8] gave an algorithm that finds a basis B of M such that $Ax_B = b$ in time $O(m\Delta)^{O(m^2)}\operatorname{poly}(n)$, where x_B is the incidence vector of B. This generalizes Binary Integer Linear Programming, since one can take E as the set of binary variables and n additional dummy variables and then take M as the uniform matroid of rank n.

Finally, Closest String (with arbitrary alphabet) can still be solved in time $m^{O(m^2)}\operatorname{poly}(n)$ for example by casting it as the previous matroid problem [8] or as a block structured ILP, see [13].

1.2 Algorithm for few distinct variables

We show that the running time of $2^{\tilde{O}(m^2)}\operatorname{poly}(n)$ for Integer Linear Programming with $\Delta = O(1)$, i.e., the subject of Hypothesis 1, can be improved if the number of distinct variables is low. For the sake of generality, we state the algorithmic result for the optimization variant and any range of Δ.

▶ **Theorem 6.** *Consider the integer programming problem*

$$\begin{aligned} &\max\ c^\mathsf{T} x \\ &\quad Ax = b \\ &\ell_i \le x_i \le u_i && i = 1, \dots, n \\ &\quad x \in \mathbb{Z}^n\ . \end{aligned} \tag{1}$$

Let Δ be an upper bound on the absolute value of entries of A. The problem (1) can be solved in time

$$n^{m+1} \cdot O(m\Delta)^m \cdot \log(\|u - \ell\|_\infty).$$

Using standard reductions, see e.g. Section 2.5.2, one may reduce $\|u - \ell\|_\infty$ to $(m\Delta)^{O(m)}$, making the logarithmic factor insignificant.

We will now interpret this running time and point out interesting special cases.

Binary ILP with $A \in \{0,1\}^{m\times n}$

Here, the running time above implies an $n'^{O(m)} + O(nm)$ time algorithm, where n' is the number of distinct variables, i.e., variables that differ either in their entry of c or in their column of A: one can merge identical variables in time $2^{O(m)} + O(nm)$ time, after which

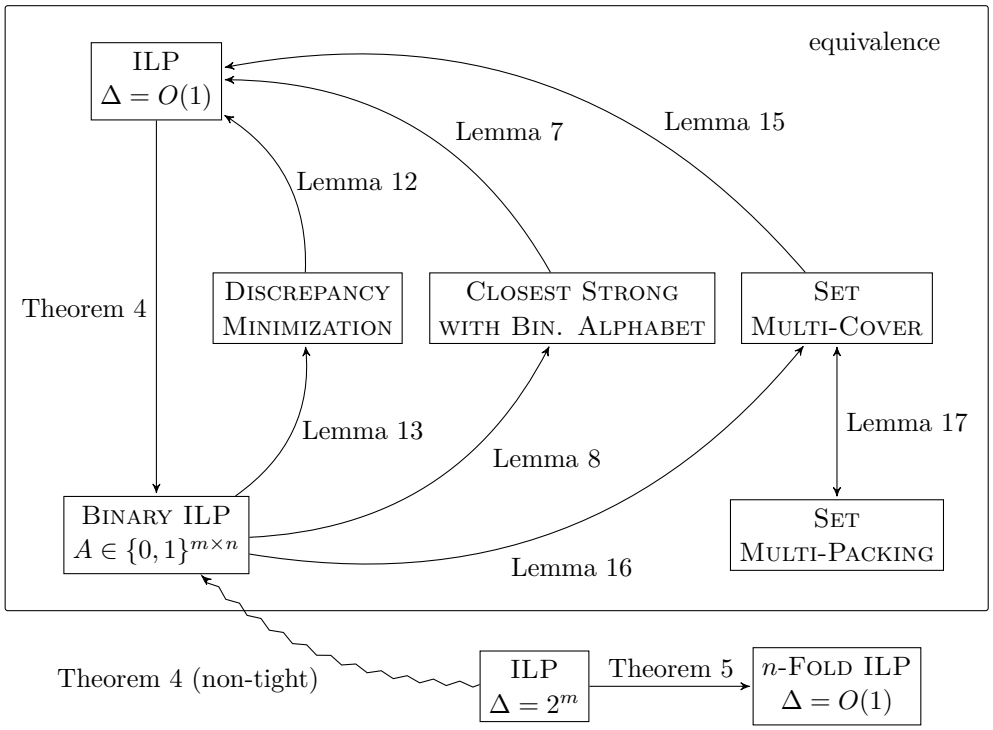


Figure 1 Overview of reductions in this paper.

$\|u\|_\infty \leq n$. Furthermore, without loss of generality, the rows of A are linearly independent, thus $m \leq n'$. Hence, the overall running time is

$$\begin{aligned} 2^{O(m)} + O(nm) + {n'}^{m+1} \cdot O(m)^m \cdot \log n &\leq {n'}^{O(m)} \cdot \log n + O(nm) \\ &\leq {n'}^{O(m)} + O(nm). \end{aligned}$$

Here, the last inequality follows from a case distinction whether $\log n$ is greater than ${n'}^{O(m)}$ or not. Note that in the setting without objective function we have $n' \leq 2^m$. Thus, this running time is at least as good as the known $2^{\tilde{O}(m^2)} \operatorname{poly}(n)$ time algorithms. It also dominates the running time of $n^{O(m)}$ one would get by simple dynamic programming over the right-hand sides b' that are feasible (for which there can be at most n^m).

Binary ILP with $A \in \{0,1\}^{m \times n}$ and a constant number of 1s in each column

Here, the number of distinct columns is polynomial in m and the previous case implies a running time of $m^{O(m)} + O(nm)$, meaning that Hypothesis 1 does not extend to this special case.

2 Reductions

The basic idea of all reductions for Theorem 2 is that we transform one problem with parameter m into another problem with parameter $m' = O(m \cdot \log^{O(1)} m)$. Additionally the size of the instance may only increase from n to $2^{O(m^{2-\delta})} \operatorname{poly}(n)$ for some $\delta > 0$. The concrete reductions we prove can be seen in Figure 1.

2.1 Closest String

Let d be the bound on the maximum hamming distance in the decision version of Closest String with Binary Alphabet. For a string $s \in \Sigma^n$ we denote by $s[i]$ the ith character. For $i \leq j$ we write $s[i \dots j]$ for the substring from the ith to the jth character.

▶ **Lemma 7.** *Theorem 2, Statement 1 implies Statement 3*

Proof. The following is an ILP model for Closest String with Binary Alphabet.

$$\sum_{i=1}^{n} x_i \cdot \mathbb{1}_{\{s_j[i]=0\}} + (1-x_i) \cdot \mathbb{1}_{\{s_j[i]=1\}} \leq d \qquad \text{for all } j \in \{1,2,\dots,m\}$$

$$x \in \{0,1\}^n$$

One may add slack variables to turn the inequalities into equalities. ◀

▶ **Lemma 8.** *Theorem 2, Statement 3 implies Statement 2*

Proof. We want to transform the following ILP

$$\begin{aligned} Ax &= b \\ x &\in \{0,1\}^n \end{aligned} \tag{2}$$

where $A \in \{0,1\}^{m \times n}$, into an equivalent instance of Closest String. We first rewrite (2) into a more convenient form.

▷ Claim 9. One can in polynomial time construct a matrix $C \in \{-1,1\}^{(2m+2)\times 2n}$ and some $c \in \mathbb{Z}^{2m+2}$ such that (2) is feasible if and only if there is a solution to

$$\begin{aligned} Cx &\leq c \\ x &\in \{0,1\}^{2n}. \end{aligned} \tag{3}$$

Furthermore, every feasible x for (3) has $x_1 + \dots + x_{2n} = n$.

This follows from simple transformations. We defer the proof until later and first show how the reduction follows from it. By comparing to the ILP formulation of Closest String we observe that (3) corresponds to a "non-uniform" variant of Closest String. It can be reformulated as: given strings $s_1, \dots, s_{2m+2} \in \{0,1\}^{2n}$ and bounds $d_1, \dots, d_{2m+2} \in \mathbb{Z}$, find a string $t \in \{0,1\}^{2n}$ such that for each $j = 1, \dots, m$ we have $d(t, s_j) \leq d_j$. This follows from the ILP model given in the proof of Lemma 7. Furthermore, we have the guarantee that any solution has exactly n ones. To transform this into a regular instance, we add two more strings r_1, r_2 and to each string s_j we add $4n$ more characters, which makes a total of $6n$ characters per string. The strings of this instance $s'_1, \dots, s'_{2m+2}, r_1, r_2$ are defined as

$$\begin{aligned} r_1 &= (\ 0,\dots,0,\ 0,\dots,0,\ 0,\dots,0,\ 0,\dots,\dots,0\ ), \\ r_2 &= (\ 1,\dots,1,\ 1,\dots,1,\ 1,\dots,1,\ 0,\dots,\dots,0\ ), \\ s'_j &= (\ \underbrace{s_j}_{2n \text{ chars}},\ \underbrace{1,\dots,1}_{n \text{ chars}},\ \underbrace{0,\dots,0}_{n \text{ chars}},\ \underbrace{1,\dots,1}_{2n-d_j \text{ chars}}\ \underbrace{0,\dots,0}_{d_j \text{ chars}}\ ). \end{aligned}$$

Here, we assume without loss of generality that $d_j \in \{0,\dots,2n\}$. We claim that there is a solution to this instance with maximum Hamming distance $2n$ if and only if there is a solution to the non-uniform instance.

▷ Claim 10. If there is a string $t' \in \{0,1\}^{6n}$ with distance at most $2n$ to r_1, r_2, and s'_j, $j = 1, 2, \ldots, 2m+2$, then there is also a string $t \in \{0,1\}^{2n}$ with distance at most d_j to s_j, $j = 1, 2, \ldots, 2m+2$.

▷ Claim 11. If there is a string $t \in \{0,1\}^{2n}$ with distance at most d_j to s_j, $j = 1, 2, \ldots, 2m+2$, then there is also a string $t' \in \{0,1\}^{6n}$ with distance at most $2n$ to r_1, r_2, and s'_j, $j = 1, 2, \ldots, 2m+2$.

From these claims, the lemma follows immediately. ◀

Proof of Claim 9. We add variables $\bar{x}_1, \ldots, \bar{x}_n$ and force a solution to take exactly n many ones

$$\begin{aligned} Ax &= b \\ x_1 + \cdots + x_n + \bar{x}_1 + \cdots + \bar{x}_n &= n \\ x, \bar{x} &\in \{0,1\}^n. \end{aligned}$$

Next, we change the equality constraints into inequalities and A into a $-1, 1$ matrix

$$\begin{aligned} A'x - \mathbf{1}\bar{x} &\leq b' \\ -A'x + \mathbf{1}\bar{x} &\leq -b' \\ x_1 + \cdots + x_n + \bar{x}_1 + \cdots + \bar{x}_n &\leq n \\ -x_1 - \cdots - x_n - \bar{x}_1 - \cdots - \bar{x}_n &\leq -n \\ x, \bar{x} &\in \{0,1\}^n \end{aligned}$$

where $A' = 2A - \mathbf{1}$ with $\mathbf{1}$ being the all-ones $m \times n$ matrix and $b' = 2b - (n, \ldots, n)^\mathsf{T}$. ◁

Proof of Claim 10. Suppose there is a string $t' \in \{0,1\}^{6n}$ with distance at most $2n$ to each of the strings $s'_1, \ldots, s'_{2m+2}, r_1, r_2$. Because $d(r_1, r_2) = 4n$, string t' must match r_1 and r_2 on characters where r_1 and r_2 match. More precisely, $t'[i] = 0$ for $i \in \{4n, \ldots, 6n\}$. Formally, this follows because of

$$\begin{aligned} 4n &\geq d(t', r_1) + d(t', r_2) \\ &= d(t'[1 \ldots 4n], r_1[1 \ldots 4n]) + d(t'[1 \ldots 4n], r_2[1 \ldots 4n]) \\ &\quad + 2d(t'[4n+1 \ldots 6n], (0, \ldots, 0)) \\ &\geq d(r_1[1 \ldots 4n], r_2[1 \ldots 4n]) + 2d(t'[4n+1 \ldots 6n], (0, \ldots, 0)) \\ &\geq 4n + 2d(t'[4n+1 \ldots 6n], (0, \ldots, 0)). \end{aligned}$$

Let us now analyze the distance between $t := t'[1 \ldots 2n]$ and s_j for some j. Since the last $2n$ characters of t' are zero, we have $d(t'[4n+1 \ldots 6n], s'_j[4n+1 \ldots 6n]) = 2n - d_j$. Thus,

$$\begin{aligned} d(t, s_j) &\leq d(t', s'_j) - d(t'[4n+1 \ldots 6n], s'_j[4n+1 \ldots 6n]) \\ &\leq 2n - (2n - d_j) = d_j. \end{aligned}$$

Thus, string t is a solution for the non-uniform instance. ◁

Proof of Claim 11. Let $t \in \{0,1\}^{2n}$ with $d(t[1 \ldots 2n], s_j) \leq d_j$ for all j. We extend it to a string $t' \in \{0,1\}^{6n}$ by setting $t'[1 \ldots 2n] = t$, $t[2n+1 \ldots 3n] = (1, \ldots, 1)$, and $t[3n+1 \ldots 6n] = (0, \ldots, 0)$. Let us now verify that t' has a distance at most $2n$ to each string. For r_1, r_2 note that $t[1 \ldots 2n]$ has exactly n ones by guarantee of the non-uniform instance. Thus, the distance to r_1 and r_2 is exactly $2n$. Consider some $j \in \{1, \ldots, 2m+2\}$. Then

$$d(s'_j, t) = d(s_j, t[1 \ldots 2n]) + 2n - d_j \leq 2n.$$ ◀

2.2 Discrepancy Minimization

▶ **Lemma 12.** *Theorem 2, Statement 1 implies Statement 4*

Proof. Let d be a bound on the objective in the decision version of Discrepancy Minimization. Let A be the incidence matrix of the given set system. Then the colorings of discrepancy at most d are exactly the solutions of

$$\begin{pmatrix} A \\ -A \end{pmatrix} y \leq \begin{pmatrix} d \\ d \\ \vdots \\ d \end{pmatrix}$$
$$y \in \{-1,1\}^n.$$

This can be equivalently formulated as

$$\begin{pmatrix} 2A \\ -2A \end{pmatrix} x \leq \begin{pmatrix} A \\ -A \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} + \begin{pmatrix} 2d \\ 2d \\ \vdots \\ 2d \end{pmatrix}$$
$$x \in \{0,1\}^n.$$

where $x_i = (1+y_i)/2$. One may translate the inequalities into equalities by introducing slack variables. Therefore, an algorithm for Integer Linear Programming can be used to solve Discrepancy Minimization. ◀

▶ **Lemma 13.** *Theorem 2, Statement 4 implies Statement 2.*

Proof. Consider an ILP of the form

$$Ax = b \tag{4}$$
$$x \in \{0,1\}^n$$

for $A \in \{0,1\}^{m \times n}$ and $n \leq m^{O(m)}$. We will construct an instance of Discrepancy Minimization which has discrepancy zero if and only if the ILP has a feasible solution. Towards this, we first reformulate the ILP above as

$$Ay = b' \tag{5}$$
$$y \in \{-1,1\}^n$$

where $b' = 2b - A\,(1,\dots,1)^\mathsf{T}$. Note that x is feasible for (4) if and only if $y = 2x - (1,\dots,1)^\mathsf{T}$ is feasible for (5). Also, if $b' = (0,\dots,0)^\mathsf{T}$, then (5) is already equivalent to an instance of Discrepancy Minimization that tests for discrepancy zero. To handle the general case, we transform it into an equivalent system with right-hand size $(0,\dots,0)^\mathsf{T}$. We first construct a gadget of elements that have the same color.

▷ Claim 14. For any $k \in \mathbb{N}$ we can construct a pair of matrices $B, \bar{B} \in \{0,1\}^{(2k-1)\times 2^k}$ such that there are exactly two solutions to

$$Bz + \bar{B}\bar{z} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$
$$z, \bar{z} \in \{-1,1\}^{2^k}$$

namely $z = (1,\dots,1)^\mathsf{T}, \bar{z} = (-1,\dots,-1)^\mathsf{T}$ and $z = (-1,\dots,-1)^\mathsf{T}, \bar{z} = (1,\dots,1)^\mathsf{T}$.

We will defer the proof to the end of the section. Using this gadget with $k = \lceil \log_2 n \rceil = O(m \log m)$ we now replace each coefficient b'_j in the previous system by the variables from the gadget. Note that (5) is infeasible if $\|b'\|_\infty > n$. Thus assume without loss of generality that $\|b'\|_\infty \leq n \leq 2^k$. Let $C, \bar{C} \in \{0,1\}^{2^k \times m}$ be defined as follows. The jth row of C has b'_j many ones at arbitrary positions if $b'_j \geq 0$ and is all zero otherwise; contrary, the jth row of $\bar{C}$ has $-b'_j$ many ones at arbitrary positions if $b'_j < 0$ and is all zero otherwise.

Now consider the system

$$\begin{aligned} Ay + Cz + \bar{C}\bar{z} &= \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \\ Bz + \bar{B}\bar{z} &= \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \\ y &\in \{-1,1\}^n \\ z, \bar{z} &\in \{-1,1\}^{2^k}. \end{aligned} \tag{6}$$

We claim that (6) has a solution if and only if there is a solution to (5). Let $y, z, \bar{z}$ be a solution to the former. Notice that the negation of a solution is also feasible. Due to Claim 14 we may assume without loss of generality that $z = (-1, \ldots, -1)^\mathsf{T}$ and $\bar{z} = (1, \ldots, 1)^\mathsf{T}$, negating the solution if necessary. It follows that

$$Cz + \bar{C}\bar{z} = -b'.$$

Thus, $Ay = b'$, which concludes the first direction. For the other direction, assume that there is a solution y to (5). We set $z = (-1, \ldots, -1)^\mathsf{T}$, $\bar{z} = (1, \ldots, 1)^\mathsf{T}$, which by Claim 14 satisfies $Bz + \bar{B}\bar{z} = (0, \ldots, 0)^\mathsf{T}$. As before we have that $Cz + \bar{C}\bar{z} = -b'$. Thus, $y, z, \bar{z}$ is a solution to (6). This establishes the equivalence of the initial ILP instance to (6), which corresponds to an instance of Discrepancy Minimization where we test for discrepancy zero with $m' = O(m \log m)$ sets. ◀

Proof of Claim 14. The existence of such a matrix can be proven by induction: for $k = 1$, we simply take $B = \bar{B} = (1)$. Now suppose that we already have a pair of matrices $B, \bar{B} \in \{0,1\}^{(2k-1)\times 2^k}$ as above. Then we set

$$B' = \begin{pmatrix} B & 0 \\ 1\cdots 1 & 0\cdots 0 \\ 0\cdots 0 & 1\cdots 1 \end{pmatrix} \text{ and } \bar{B}' = \begin{pmatrix} \bar{B} & 0 \\ 0\cdots 0 & 1\cdots 1 \\ 1\cdots 1 & 0\cdots 0 \end{pmatrix} \in \{0,1\}^{(2k+1)\times 2\cdot 2^k}.$$

It can easily be checked that choosing either $z = z' = (1, \ldots, 1)^\mathsf{T}, \bar{z} = \bar{z}' = (-1, \ldots, -1)^\mathsf{T}$ or $z = z' = (-1, \ldots, -1)^\mathsf{T}, \bar{z} = \bar{z}' = (1, \ldots, 1)^\mathsf{T}$ satisfies

$$B' \begin{pmatrix} z \\ z' \end{pmatrix} + \bar{B}' \begin{pmatrix} \bar{z} \\ \bar{z}' \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{7}$$

Now take any $z, z', \bar{z}, \bar{z}' \in \{-1,1\}^{2^{k+1}}$ that satisfy (7). Then we have that $Bz + \bar{B}\bar{z} = (0, \ldots, 0)^\mathsf{T}$. Hence by induction hypothesis either $z = (1, \ldots, 1)^\mathsf{T}, \bar{z} = (-1, \ldots, -1)^\mathsf{T}$ or $z = (-1, \ldots, -1)^\mathsf{T}, \bar{z} = (1, \ldots, 1)^\mathsf{T}$. Assume for now the first case holds. Then because of the second-to-last rows of $B, \bar{B}$ we have

$$2^k + \sum_{i=1}^{2^k} \bar{z}'_i = \sum_{i=1}^{2^k} z_i + \bar{z}'_i = 0.$$

Hence, $\bar{z}' = (-1, \dots, -1)^\mathsf{T} = \bar{z}$. Similarly, the last row of $B, \bar{B}$ implies that $z' = (1, \dots, 1)^\mathsf{T} = z$. Analogously, if $z = (-1, \dots, -1)^\mathsf{T}, \bar{z} = (1, \dots, 1)^\mathsf{T}$ then $z' = z$ and $\bar{z}' = \bar{z}$. ◁

2.3 Set Multi-Cover

▶ **Lemma 15.** *Theorem 2, Statement 1 implies Statement 5.*

Proof. An instance of the decision version of Set Multi-Cover with a bound d on the cardinality can be formulated as

$$\begin{aligned} x_1 + \cdots + x_n &\leq d \\ Ax &\geq \begin{pmatrix} b \\ \vdots \\ b \end{pmatrix} \\ x &\in \{0,1\}^n \end{aligned}$$

where $A \in \{0,1\}^{m \times n}$ is the incidence matrix of the given set system. This can easily be translated to the form of (1) by introducing slack variables. Notice that this ILP has $m + 1$ constraints. Thus, a faster algorithm for ILP would imply a faster algorithm for Set Multi-Cover. ◀

In the remainder, we show that the converse is also true.

▶ **Lemma 16.** *Theorem 2, Statement 5 implies Statement 2.*

Proof. First, we reduce to a "non-uniform" version of Set Multi-Cover where each element v has a different demand b_v. Let $A \in \{0,1\}^{m \times n}$ and $b \in \mathbb{Z}^m_{\geq 0}$ and consider the solutions to

$$\begin{aligned} Ax &= b \\ x &\in \{0,1\}^n. \end{aligned}$$

First, we add n additional binary variables $\bar{x}$ and the requirement that exactly n variables are equal to one, i.e.,

$$\begin{aligned} Ax &= b \\ x_1 + \cdots + x_n + \bar{x}_1 + \cdots + \bar{x}_n &= n \\ x &\in \{0,1\}^n \\ \bar{x} &\in \{0,1\}^n. \end{aligned}$$

This system is equivalent to the previous one, by setting $n - x_1 - \cdots - x_n$ arbitrary variables $\bar{x}_i$ to 1. Next, we transform the equality constraints $Ax = b$ into inequalities:

$$
\begin{aligned}
Ax &\geq b \\
(\mathbf{1} - A)x + \mathbf{1}\bar{x} &\geq \begin{pmatrix} n \\ \vdots \\ n \end{pmatrix} - b \\
x_1 + \cdots + x_n + \bar{x}_1 + \cdots + \bar{x}_n &= n \\
x &\in \{0,1\}^n \\
\bar{x} &\in \{0,1\}^n.
\end{aligned}
$$

Here, $\mathbf{1}$ denotes the $n \times n$ all-ones matrix. Note that the second constraint is equivalent to $Ax \leq b$, since we fixed the number of ones in the solution. This ILP is feasible if and only if the optimum of the following ILP is n.

$$
\begin{aligned}
\min\ & x_1 + \cdots + x_n + \bar{x}_1 + \cdots + \bar{x}_n \\
& Ax \geq b \\
& (\mathbf{1} - A)x + \mathbf{1}\bar{x} \geq \begin{pmatrix} n \\ \vdots \\ n \end{pmatrix} - b \\
& x_1 + \cdots + x_n + \bar{x}_1 + \cdots + \bar{x}_n \geq n \\
& x \in \{0,1\}^n \\
& \bar{x} \in \{0,1\}^n.
\end{aligned}
$$

This ILP corresponds to an instance of non-uniform Set Multi-Cover with $2m+1$ elements.

To reduce a non-uniform instance $S_1, \ldots, S_n$, $(b_v)_{v \in U}$, to a uniform instance of Set Multi-Cover we proceed as follows: add one new element and n many new sets to the instance. The coverage requirement of each element is n. The new element is contained in each of the new sets and in none of the old ones. Thus, each new set has to be taken. Furthermore, we add each old element v to $n - b_v$ many arbitrary new sets. ◀

2.4 Set Multi-Packing

▶ **Lemma 17.** *Theorem 2, Statements 5 and 6 are equivalent.*

Proof. Notice the following duality between Set Multi-Cover and Set Multi-Packing. Let $U = \{1, 2, \ldots, m\}$, $S_1, S_2, \ldots, S_n$, and $b \in \mathbb{N}$ be an instance of Set Multi-Cover. Now consider the instance of Set Multi-Packing with universe U, set system $\bar{S}_1 = U \setminus S_1, \bar{S}_2 = U \setminus S_2, \ldots, \bar{S}_n = U \setminus S_n$, and bounds $\bar{b} = n - b$. This is a bipartition between instances of Set Multi-Cover and Set Multi-Packing, i.e., it can be performed in both ways.

For one pair of such instances, a solution I for Set Multi-Cover is feasible if and only if $\bar{I} = \{1, 2, \ldots, n\} \setminus I$ is feasible for Set Multi-Packing. Thus, if the optimum of Set Multi-Cover is k, then the optimum of Set Multi-Packing is $n - k$. ◀

2.5 Integer Linear Programming

In this section, we prove Theorem 4, i.e., we show how to reduce an ILP with large coefficients into a (larger) ILP with only 0/1 coefficients.

▶ **Theorem 4.** *There is a polynomial time algorithm that transforms an instance of* Integer Linear Programming *with* $\Delta > 1$ *into an equivalent one with* $A' \in \{0,1\}^{m' \times n'}$ *for* $m' = O(m \log \Delta)$ *and* $n' \leq {m'}^{O(m')}$.

Furthermore, we show how to reduce an ILP with arbitrary upper and lower bounds into one with at most $(m\Delta)^{O(m)}$ binary variables. Note that this implies that Statements 1 and 2 are equivalent and concludes the proof of Theorem 2.

2.5.1 From large coefficients to zero-one

We will transform an ILP of the form

$$\begin{aligned} Ax &= b \\ \ell \leq x &\leq u \\ x &\in \mathbb{Z}^n \end{aligned}$$

where $A \in \{-\Delta, \dots, \Delta\}^{m \times n}$ into an equivalent one with $A' \in \{0,1\}^{m' \times n'}$ for $m' = O(m \log \Delta)$. Let $k = \lceil \log_2(\Delta) \rceil$. For the jth row of A we introduce $4(k+1)$ rows in A', denoted by $r_0^+(j), {r'}_0^+(j), \dots, r_k^+(j), {r'}_k^+(j), r_0^-(j), {r'}_0^-(j), \dots, r_k^-(j), {r'}_k^-(j)$. Intuitively, the rows $r_i^+(j), {r'}_i^+(j)$ are symmetric rows that each stand for a value of 2^i in row j. Similarly, $r_i^-(j), {r'}_i^-(j)$ stand for a value of -2^i in row j. The right-hand sides of the new ILP are b_j for row $r_0^+(j)$ and zero for all other rows affiliated with j.

For each column A_i we derive a column of A' as follows: for row j we consider the binary encoding of A_{ji} and have the column in A' use this binary encoding in $r_0^+(j), r_1^+(j), \dots, r_k^+(j)$ if $A_{ji} \geq 0$ and in $r_0^-(j), r_1^-(j), \dots, r_k^-(j)$ if $A_{ji} < 0$. All other entries of this column are zero.

We now add auxiliary variables to shift the values from one power to another. For each row j and each $i = 0, \dots, k-1$ we add:

- a variable $x(i,j)$ with a one in row $r_i^+(j), {r'}_i^+(j)$ and $r_{i+1}^-(j)$ and
- a variable $\bar{x}(i,j)$ with a one in row $r_i^-(j), {r'}_i^-(j)$ and $r_{i+1}^+(j)$.

Furthermore, for each row j and each $i = 0, \dots, k$ we add:

- a variable $x_a(i,j)$ with a one in row ${r'}_i^+(j)$ and $r_i^-(j)$,
- a variable $x_b(i,j)$ with a one in row ${r'}_i^-(j)$ and $r_i^+(j)$, and
- a variable $x_c(i,j)$ with a one in row $r_i^-(j)$ and $r_i^+(j)$.

Note that each auxiliary variable does not alter the total value for row j, i.e., the sum of 2^i times $r_i^+(j)$ and ${r'}_i^+(j)$ minus 2^i times $r_i^-(j)$ and ${r'}_i^-(j)$ across all i.

Thus, any solution to the new ILP must correspond to a solution of the original ILP. The converse is also true: the auxiliary variables can be adjusted to preserve the original value of the ILP. This is because any value for one of the rows of j can always be shifted to $r_0^+(j)$ using the auxiliary variables, while still satisfying the constraints. Therefore, we can enforce the value of the unchanged variables.

2.5.2 From bounded to binary variables

Consider an ILP of the form

$$\begin{aligned} Ax &= b \\ \ell \leq x &\leq u \\ x &\in \mathbb{Z}^n \end{aligned}$$

where $A \in \{-\Delta, \dots, \Delta\}^{m \times n}$. We will first transform this into an equivalent ILP of the form

$$\begin{aligned} A'x &= b' \\ x &\in \{0,1\}^{n'} \end{aligned} \tag{8}$$

where $A' \in \{-\Delta, \dots, \Delta\}^{m \times n'}$ for $n' \leq (m\Delta)^{O(m)}$. Assume without loss of generality that each column of A is different. Otherwise, we can merge identical columns together by adding the respective upper and lower bounds. This implies that $n \leq (2\Delta+1)^m$. Next, we compute a vertex solution x^* to the continuous relaxation $\{Ax = b, \ell \leq x \leq u, x \in \mathbb{R}^n\}$. The proximity bound by Eisenbrand and Weismantel [9] shows that there is an integer solution z with $\|z - x^*\|_1 \leq m(2m\Delta+1)^m$ if any integer solution exists. Thus, choosing $\ell'_i = \max\{\ell_i, \lceil x^*_i \rceil - (2\Delta+1)^m\}$ and $u'_i = \min\{u_i, \lfloor x^*_i \rfloor + (2\Delta+1)^m\}$ for $i = 1, 2, \dots, n$, we can replace ℓ, u by ℓ', u' without affecting feasibility. By shifting the solution, we can reduce the lower bound to zero and we arrive at the following ILP that is equivalent to the original one.

$$\begin{aligned} Ax &= b - A\ell' \\ x_i &\in \{0, 1, \dots, u'_i - \ell'_i\} \qquad\qquad \text{for all } i = 1, 2, \dots, n \end{aligned}$$

Note that $u'_i - \ell'_i \leq 2m(2m\Delta+1)^m$. Thus replacing each variable by $u'_i - \ell'_i$ binary variables we arrive at an ILP with $n' \leq 2m(2m\Delta+1)^m \cdot n \leq 2m(2m\Delta+1)^{2m}$ binary variables.

2.6 n-Fold Integer Linear Programming

In this section, we prove Theorem 5.

▶ **Theorem 5.** *For every $\delta > 0$, there is no algorithm with running time $2^{O(m^{3-\delta})}\operatorname{poly}(n)$ for n-Fold Integer Linear Programming when the maximum absolute entry is bounded by $\Delta = O(1)$, unless Hypothesis 3 is false.*

Consider the ILP

$$\begin{aligned} Ax &= b \\ x &\in \{0,1\}^n \end{aligned} \tag{9}$$

with $A \in \{-2^m, \dots, 2^m\}^{m \times n}$. We will show that this can be formulated as an equivalent n-Fold Integer Linear Program with parameter m and $\Delta = O(1)$. The reduction follows a similar idea as one in [10], which derives a lower bound based on Subset Sum. Note that if (9) had arbitrary variables (not necessarily binary) then we can use the reduction of the previous section to transform it into a binary one. For $m \geq 3$, define

$$B = \begin{pmatrix} 1 & 0 & & \cdots & 0 & 1 \\ 0 & & & \cdots & & 0 \\ 2 & -1 & & \cdots & & \\ & 2 & -1 & \cdots & & \vdots \\ & & & \ddots & & \\ & & & 2 & -1 & 0 \end{pmatrix} \in \{-1, 0, 1, 2\}^{m \times m}.$$

This matrix has the property that the system $Bx = (1, 0, \dots, 0)^\mathsf{T}, x \in \mathbb{Z}^m_{\geq 0}$ has exactly two solutions, namely $x = (0, \dots, 0, 1)$ and $x = (1, 2^1, 2^2, \dots, 2^{m-2}, 0)$. Our n-Fold Integer Linear Program has one block A'_i, B'_i for each column A_i of A. Matrix B'_i is defined as B above with right-hand side $b'^{(i)} = (1, 0, \dots, 0)^\mathsf{T}$. Matrix A'_i is derived from A_i as follows: consider a coefficient A_{ji}. We rewrite

$$A_{ji} = \lambda_0 \cdot 2^0 + \lambda_1 \cdot 2^1 + \cdots + \lambda_{m-2} \cdot 2^{m-2} \text{ with } \lambda \in \{-2, \dots, 2\}^{m-1}.$$

Such a λ exists since $|A_{ji}| \leq 2^m$. Then we set the jth row of A'_i as $(\lambda^\mathsf{T}, 0)$. Let $x^{(i)}$ be the variables corresponding to block A'_i, B'_i. By choice of $B'_i, b'^{(i)}$ there are exactly two settings of $x^{(i)}$ that satisfy $B'_i x^{(i)} = b'^{(i)}$, namely $x^{(i)} = (0, \dots, 0, 1)$ and $x^{(i)} = (2^0, 2^1, \dots, 2^{m-2}, 0)$. Thus $A'_i x^{(i)} \in \{(0, \dots, 0)^\mathsf{T}, A_i\}$. Hence, the n-Fold Integer Linear Program with $b^{(0)} = b$ is equivalent to (9).

3 Algorithm for few distinct variables

In this section, we prove Theorem 6.

▶ **Theorem 6.** *Consider the integer programming problem*

$$\begin{aligned} &\max\ c^\mathsf{T} x \\ &\quad Ax = b \\ &\ell_i \leq x_i \leq u_i \qquad\qquad i = 1, \dots, n \\ &\quad x \in \mathbb{Z}^n\ . \end{aligned} \tag{1}$$

Let Δ be an upper bound on the absolute value of entries of A. The problem (1) can be solved in time

$$n^{m+1} \cdot O(m\Delta)^m \cdot \log(\|u - \ell\|_\infty).$$

We assume without loss of generality that in (1) we have $\ell = (0, 0, \dots, 0)$. Note that otherwise, one may obtain an equivalent ILP with $\ell = (0, 0, \dots, 0)$, $u' = u - \ell$, and $b' = b - A\ell$. We first decompose each u_i into a sum of powers of two with the properties described in the following lemma. Such a construction is well-known in literature, see e.g. [18, 4, 16].

▶ **Lemma 18** (Cover). *For every integer $k \in \mathbb{N}$ and $h \geq \lfloor \log k \rfloor$, there exist integers $c_0(k), \dots, c_h(k) \in \{0, 1, 2\}$ such that*

$$\left\{ \sum_{i=0}^{h} 2^i x_i : 0 \leq x_i \leq c_i(k) \text{ for all } 0 \leq i \leq h \right\} = \{0, 1, \dots, k\}.$$

We call such a set $c_0(k), \dots, c_h(k)$ a cover *of k. This set can be found in polynomial time in h.*

Proof. For $n \in \mathbb{N}$, let $\mathrm{bin}_i(n) \in \{0, 1\}$ denote the ith bit of the binary representation of n. Let $B := 2^h - 1$ be the *bottom* bits, and let $T = k - B$ represent the *top* bits of k. Now, consider the sets

$$S_B := \left\{ \sum_{i=0}^{h-1} 2^i x_i : 0 \leq x_i \leq \mathrm{bin}_i(B) \text{ for all } 0 \leq i \leq h-1 \right\} \text{ and}$$

$$S_T := \left\{ \sum_{i=0}^{h} 2^i x_i : 0 \leq x_i \leq \mathrm{bin}_i(T) \text{ for all } 0 \leq i \leq h \right\}.$$

We will set $c_i(k) := \mathrm{bin}_i(B) + \mathrm{bin}_i(T) \in \{0,1,2\}$. Showing that these numbers are a cover of k is equivalent to

$$S_B \oplus S_T := \{b + t \mid b \in S_B, t \in S_T\} = \{0, 1, \ldots, k\}.$$

First, observe that $S_B = \{0, 1, \ldots, 2^h - 1\}$. Moreover, there is no gap of length at least 2^h in the set S_T, that is, for every nonzero $a \in S_T$, there exists a smaller $b \in S_T$ such that $a - b < 2^h$. This b can be constructed by flipping an arbitrary bit of a to 0. Therefore, $S_B \oplus S_T$ consists of consecutive numbers. The smallest number in $S_B \oplus S_T$ is clearly 0, and the largest is $B + T = k$. ◀

We set $h = \lfloor \log_2(\|u\|_\infty) \rfloor$. Using the lemma above, for each i, we decompose $u_i = c_0(u_i) \cdot 2^0 + c_1(u_i) \cdot 2^1 + \cdots + c_h(u_i) \cdot 2^h$ with its cover. We can now naturally rewrite (1) as

$$\begin{aligned}
&\max \sum_{i=0}^{h} 2^i c^\mathsf{T} z^{(i)} \\
&\sum_{i=0}^{h} 2^i A z^{(i)} = b \\
&0 \le z_i^{(k)} \le c_k(u_i) && k \in \{0, \ldots, h\} \\
&z^{(0)}, \ldots, z^{(h)} \in \mathbb{Z}^n \ .
\end{aligned}$$

Each $z^{(j)}$ produces a partial solution for some unknown right-hand side $b^{(j)} = 2^j A z^{(j)}$. We write $b^{(\le j)} = 2^0 A z^{(0)} + 2^1 A z^{(1)} + \cdots + 2^j A z^{(j)}$.

Our approach is now to compute, for $j = 0, 1, \ldots, h$ and for every potential value of $b^{(\le j)}$, a solution $z^{(0)}, z^{(1)}, \ldots, z^{(j)}$. To do this, we first reduce the search space of relevant vectors $b^{(\le j)}$.

▶ **Lemma 19.** *Consider $A \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}^m$ with $\|A\|_\infty \le \Delta$. Suppose that $y = y^{(0)} \cdot 2^0 + y^{(1)} \cdot 2^1 + \cdots + y^{(h)} \cdot 2^h$ satisfies $Ay = b$ and $0 \le y^{(j)} \le u^{(j)}$ with $u^{(j)} \in \{0,1,2\}^n$ for each $j \in \{0, \ldots, \ell\}$. Then for $b^{(\le j)} = 2^0 A y^{(0)} + 2^1 A y^{(1)} + \cdots + 2^j A y^{(j)}$, we have*

$$b^{(\le j)} \equiv b \mod 2^{j+1}$$

and furthermore

$$b^{(\le j)} \in \{-2^{j+2} mn\Delta, \ldots, 2^{j+2} mn\Delta\}^m \ .$$

Proof. The first property follows from the observation that $b^{(>j)} := b - b^{(\le j)}$ is a vector with each component being a multiple of 2^{j+1}. Hence, $b^{(\le j)} \equiv b - b^{(>j)} \equiv b \mod 2^{j+1}$.

For the second property, observe that $\|y^{(h)}\|_1 \le \|u^{(h)}\|_1 \le 2n$ for each h. Therefore, $\|b^{(\le j)}\|_\infty \le \sum_{h=0}^{j} 2^{h+1} n\Delta \le 2^{j+2} nm\Delta$. ◀

We say that a vector $b' \in \mathbb{Z}^m$ is *relevant* for j if $b' \equiv b \mod 2^{j+1}$ and $\|b'\|_\infty \le 2^{j+2} mn\Delta$. Clearly, the following holds:

▷ Claim 20. For every j, the number of relevant vectors for j is $O(mn\Delta)^m$.

We will now iteratively compute solutions for all vectors relevant for $j+1$ from the solutions for all relevant vectors for j with $j \ge 0$. Note that this will immediately establish Theorem 6.

▶ **Lemma 21.** *Let $\mathcal{S}_j$ be a set of optimal solutions to (1) for all $b' \in \mathbb{Z}^m$ that are relevant for j. Then, in time $n^{m+1} \cdot O(m\Delta)^m$, we can compute a set $\mathcal{S}_{j+1}$ of optimal solutions for all $b'' \in \mathbb{Z}^m$ relevant for $j+1$.*

Proof. Let V be the set of all vectors $b' \in \mathbb{Z}^m$ with $b' \equiv b \mod 2^{j+1}$ and $\|b'\|_\infty \leq 2^{j+2}mn\Delta$.

We define an edge-weighted directed acyclic graph with vertices $\{s\} \cup V^{(0)} \cup V^{(1)} \cup \ldots \cup V^{(n)}$, where $V^{(0)}, V^{(1)}, \ldots, V^{(n)}$ are $n+1$ copies of V, and s is a distinguished source vertex.

If $j \geq 0$, there is an edge from s to every vertex $v_{b'} \in V^{(0)}$ such that the vector $v_{b'}$ corresponds to a relevant vector $b' \in \mathbb{Z}^m$ for j for which (1) has a solution $x_{b'}$. This edge indicates feasibility, and the weight of the edge from s to $v_{b'}$ is the value $c^\mathsf{T} x_{b'}$ for the optimal solution $x_{b'}$. In the base case where $j < 0$, there is exactly one edge of weight 0 to the vertex in $V^{(0)}$ that corresponds to the all-zero vector.

For each vertex in $V^{(i-1)}$ for $0 < i \leq n$, there is an edge to the corresponding vertex in $V^{(i)}$ with weight zero. Further, if $c_{j+1}(u_i) \geq 1$, for every vertex corresponding to b' in $V^{(i-1)}$, there is an edge to the vertex in $V^{(i)}$ that corresponds to $b' + 2^j A_i$ (if it exists). The weight of this edge is $2^j c_i$.

Similarly, if $c_{j+1}(u_i) \geq 2$, then for every vertex in $V^{(i-1)}$ that corresponds to a relevant b', there is an edge to a vertex in $V^{(i)}$ that corresponds to $b' + 2^{j+1} A_i$ (if it exists), with a cost of $2^{j+1} c_i$.

Finally, we compute the longest path from s to each vertex in $V^{(n)}$, and we store these as the values of the solutions to all the relevant right-hand sides for $j+1$.

For the running time, observe that by Claim 20, we have $|V| \leq O(mn\Delta)^m$. Hence, the graph has $n^{m+1} \cdot O(m\Delta)^m$ vertices and edges. Thus, the longest path problem can be solved in time $n^{m+1} \cdot O(m\Delta)^m$.

For correctness, consider a path in the graph from vertex $v_{b_1} \in V^{(0)}$ to vertex $v_{b_2} \in V^{(n)}$ (corresponding to vectors b_1 and b_2 respectively). The edges of this path define a vector $z^{(j+1)}$ such that $0 \leq z^{(j+1)} \leq c_{j+1}(u)$. Moreover, by construction, it holds that $2^j A z^{(j+1)} + b_1 = b_2$. Finally, the weight of this path corresponds to the value $2^j c^\mathsf{T} z^{(j+1)}$. ◀

References

1. Amir Abboud, Karl Bringmann, Danny Hermelin, and Dvir Shabtay. SETH-based lower bounds for subset sum and bicriteria path. *ACM Transactions on Algorithms (TALG)*, 18(1):1–22, 2022. doi:10.1145/3450524.
2. Karl Bringmann. Fine-grained complexity theory. In *Proceedings of STACS*, page 1, 2019.
3. Moses Charikar, Alantha Newman, and Aleksandar Nikolov. Tight Hardness Results for Minimizing Discrepancy. In Dana Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011*, pages 1607–1614. SIAM, 2011. doi:10.1137/1.9781611973082.124.
4. Lin Chen, Jiayi Lian, Yuchen Mao, and Guochuan Zhang. Faster algorithms for bounded knapsack and bounded subset sum via fine-grained proximity results. In *Proceedings of SODA*, pages 4828–4848, 2024. doi:10.1137/1.9781611977912.171.
5. Jana Cslovjecsek, Friedrich Eisenbrand, Christoph Hunkenschröder, Lars Rohwedder, and Robert Weismantel. Block-structured integer and linear programming in strongly polynomial and near linear time. In *Proceedings of SODA*, pages 1666–1681, 2021. doi:10.1137/1.9781611976465.101.
6. Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström. On problems as hard as CNF-SAT. *ACM Transactions on Algorithms (TALG)*, 12(3):1–24, 2016. doi:10.1145/2925416.
7. Daniel Dadush, Arthur Léonard, Lars Rohwedder, and José Verschae. Optimizing low dimensional functions over the integers. In *Proceedings of IPCO*, pages 115–126, 2023. doi:10.1007/978-3-031-32726-1_9.
8. Friedrich Eisenbrand, Lars Rohwedder, and Karol Węgrzycki. Sensitivity, Proximity and FPT Algorithms for Exact Matroid Problems. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1610–1620, 2024. doi:10.1109/FOCS61266.2024.00100.

9 Friedrich Eisenbrand and Robert Weismantel. Proximity results and faster algorithms for integer programming using the Steinitz lemma. *ACM Transactions on Algorithms (TALG)*, 16(1):1–14, 2019. doi:10.1145/3340322.

10 Christoph Hunkenschröder, Kim-Manuel Klein, Martin Koutecký, Alexandra Lassota, and Asaf Levin. Tight lower bounds for block-structured integer programs. In *Proceedings of IPCO*, pages 224–237, 2024. doi:10.1007/978-3-031-59835-7_17.

11 Klaus Jansen and Lars Rohwedder. On integer programming, discrepancy, and convolution. *Mathematics of Operations Research*, 48(3):1481–1495, 2023. doi:10.1287/MOOR.2022.1308.

12 Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of operations research*, 12(3):415–440, 1987. doi:10.1287/MOOR.12.3.415.

13 Dušan Knop, Martin Koutecký, and Matthias Mnich. Combinatorial n-fold integer programming and applications. *Mathematical Programming*, 184(1):1–34, 2020. doi:10.1007/S10107-019-01402-2.

14 Dušan Knop, Michał Pilipczuk, and Marcin Wrochna. Tight complexity lower bounds for integer linear programming with few constraints. *ACM Transactions on Computation Theory (TOCT)*, 12(3):1–19, 2020. doi:10.1145/3397484.

15 Hendrik W. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of operations research*, 8(4):538–548, 1983. doi:10.1287/MOOR.8.4.538.

16 Silvano Martello and Paolo Toth. *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., 1990.

17 Christos H. Papadimitriou. On the complexity of integer programming. *Journal of the ACM (JACM)*, 28(4):765–768, 1981. doi:10.1145/322276.322287.

18 Adam Polak, Lars Rohwedder, and Karol Węgrzycki. Knapsack and subset sum with small items. In *Proceedings of ICALP*, pages 1–19, 2021. doi:10.4230/LIPICS.ICALP.2021.106.

19 Victor Reis and Thomas Rothvoss. The subspace flatness conjecture and faster integer programming. In *Proceedings of FOCS*, pages 974–988, 2023. doi:10.1109/FOCS57990.2023.00060.

20 Virginia Vassilevska Williams. Hardness of easy problems: Basing hardness on popular conjectures such as the strong exponential time hypothesis (invited talk). In *Proceedings of IPEC*, 2015. doi:10.4230/LIPIcs.IPEC.2015.17.

Quantum Communication Complexity of Classical Auctions

Aviad Rubinstein
Department of Computer Science, Stanford University, CA, USA

Zixin Zhou
Department of Computer Science, Stanford University, CA, USA

Abstract

We study the fundamental, classical mechanism design problem of single-buyer multi-item Bayesian revenue-maximizing auctions under the lens of communication complexity between the buyer and the seller. Specifically, we ask whether using quantum communication can be more efficient than classical communication. We have two sets of results, revealing a surprisingly rich landscape – which looks quite different from both quantum communication in non-strategic parties, and classical communication in mechanism design.

We first study the expected communication complexity of approximately optimal auctions. We give quantum auction protocols for buyers with unit-demand or combinatorial valuations that obtain an arbitrarily good approximation of the optimal revenue while running in exponentially more efficient communication compared to classical approximately optimal auctions. However, these auctions come with the caveat that they may require the seller to charge exponentially large payments from a deviating buyer. We show that this caveat is necessary - we give an exponential lower bound on the product of the expected quantum communication and the maximum payment.

We then study the worst-case communication complexity of exactly optimal auctions in an extremely simple setting: additive buyer's valuations over two items. We show the following separations:

- There exists a prior where the optimal classical auction protocol requires infinitely many bits, but a one-way message of 1 qubit and 2 classical bits suffices.
- There exists a prior where no finite one-way quantum auction protocol can obtain the optimal revenue. However, there is a barely-interactive revenue-optimal quantum auction protocol with the following simple structure: the seller prepares a pair of qubits in the EPR state, sends one of them to the buyer, and then the buyer sends 1 qubit and 2 classical bits.
- There exists a prior where no multi-round quantum auction protocol with a finite bound on communication complexity can obtain the optimal revenue.

2012 ACM Subject Classification Theory of computation → Algorithmic mechanism design; Theory of computation → Communication complexity; Theory of computation → Quantum computation theory

Keywords and phrases Mechanism design, Communication complexity, Quantum computing

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.84

Related Version *Full Version*: `https://arxiv.org/abs/2311.12444`

Funding *Aviad Rubinstein*: Supported by NSFCCF-1954927, and a David and Lucile Packard Fellowship.
Zixin Zhou: Supported by NSFCCF-1954927, and the Stanford interdisciplinary graduate fellowship.

1 Introduction

We study the quantum communication complexity of classical problems with strategic constraints. The communication problems we study differ from traditional ("cooperative") problems in communication complexity like Set Disjointness due to Incentive Compatibility


© Aviad Rubinstein and Zixin Zhou;
licensed under Creative Commons License CC-BY 4.0

(IC) constraints. Informally speaking, one can think of the goal of a traditional communication problem is to design a protocol for both parties that optimizes some common objective function.

For instance, in Set Disjointness, the common goal is to maximize the probability of outputing the correct answer. By contrast, in strategic communication, each party optimizes a different objective; we seek protocols that are communication-efficient, and at the same time don't expect strategic parties to take actions that are mis-aligned with their objectives.

Specifically, we study the quantum communication complexity of a fundamental setting in mechanism design: a monopolistic, revenue-maximizing seller with Bayesian prior auctioning n items to a single buyer; this setting has proved very attractive to researchers in theoretical computer science (e.g. [6, 18, 5, 40, 44, 70] and references therein).

It is known that even with seemingly benign Bayesian priors, revenue-optimality requires complex auctions, e.g. ones that allow the buyer to choose among lotteries [73, 55, 16, 62, 48]. This realization has sparked a fruitful line of work on the simplicity-vs-complexity of (approximately) optimal auctions. Understanding the tradeoffs of simplicity vs. complexity requires formal definitions of complexity.

Perhaps the most well-studied notion of complexity for this problem is the number of distinct lotteries offered to the buyer ("menu-size complexity" [47]). The measure exactly characterizes the *deterministic communication* complexity of the interaction between a buyer and seller who both know the rules of the auction[1] [5].

We study the communication complexity of auctions subject to an Incentive Compatibility (IC) constraint: it is crucial that a strategic buyer must not be able to gain from deviating from the protocol. (As is standard in the literature, we assume that the seller is non-strategic and follows the protocol faithfully. See Section 4 for formal definition, and e.g. [34, 29, 70, 69] for further discussion.) Under this IC constraint, [70] show that it is possible to obtain dramatic savings in communication by considering (classical) *randomized communication*. The main question we ask in this work is whether *quantum communication* can be even more efficient than classical randomized communication complexity for this problem:

Can quantum auction protocols achieve super-classical performance?

Specifically, [70] show that even though randomized auction protocols can be much more efficient than deterministic ones, they still have limitations:

- **Worst-case vs expected CC barrier:** [70] improve the communication complexity in expectation (over the randomness of the protocol), but the worst-case communication complexity of randomized protocols is still characterized by the menu-size complexity.
- **Combinatorial valuations barrier:** For buyers with combinatorial valuations over the items, [70] prove exponential lower bounds even for the expected communication of randomized auction protocols. These lower bounds hold even for approximately optimal mechanisms, and even against restricted classes of valuations (e.g. monotone submodular valuations).

In this paper, we investigate to what extent quantum communication can break these classical barriers.

[1] Specifically, deterministic-CC $= \log\big(\lceil$menu-size complexity$\rceil\big)$.

1.1 Our Contributions

We formalize a model of *quantum auction protocols* (Definition 7) that extends the randomized auction protocols of [70] by allowing the buyer and seller to send, receive, operate on, and measure qubits. We then provide two sets of results, centered around the two barriers for classical auction protocols mentioned above.

1.1.1 (Un)expected Quantum CC with Combinatorial Valuations

Our first result is an exponential quantum speed-up for auction protocols. It is stated in a general form for a mechanism that chooses an allocation among B possible allocations. Specifically, it gives a near-equivalent IC quantum auction protocol that uses only $O(\log(B))$ qubits – matching the cost of naively encoding the allocation (without strategic considerations).

▶ **Theorem 1** (Efficient in-expectation quantum auction protocols). *Let $\mathcal{D}$ be a prior over buyer's combinatorial valuations over n items; assume all valuations are in the range*[2] *$[0, U]$. Let $\mathcal{M}$ be any mechanism that can only possibly allocate one of B subsets of the items. Finally, let $\delta > 0$ be any parameter (δ may be a function of n or B). Then there is an IC quantum auction protocol that guarantees a $(1-\delta)$-fraction of $\mathcal{M}$'s expected revenue using $O(\log(B))$ qubits in expectation.*

As a corollary, we only need $O(\log(n))$ qubits for unit-demand, or $O(n)$ qubits for arbitrary combinatorial valuations. On the contrary, [70] shows that any randomized classical communication protocol requires at least $\Omega(n)$ bits for unit-demand, and $\Omega(2^n)$ bits for combinatorial valuations. To the best of our knowledge, this is the first exponential separation of quantum and classical communication in algorithmic game theory.

The positive result in Theorem 1 has a caveat: the protocol may require large payments. Specifically, we need the ability to inflict a large penalty on buyers who deviate from the intended quantum strategy. However, the probability of catching deviating buyers may be exponentially small, so we use exponentially large payments. Even though buyers who follow the protocol can never be penalized, to be accountable for a potentially large payment the buyer may need significant collateral to participate in the auction. Our second result shows that unfortunately without exponentially large payments all the lower bounds from [70] extend to quantum communication.

▶ **Theorem 2** (Efficient protocols require large payments - short version). *Let n be the number of items, P be an upper bound on the payments in the quantum auction protocol (when the valuations are normalized to $[0, 1]$), and $\hat{K}$ an upper bound on the expected communication complexity. Then for a buyer with combinatorial (XOS) valuations, any quantum auction protocol that obtains $\Omega(1)$-approximation to the optimal revenue must satisfy $\hat{K}P = 2^{\Omega(n)}$. See Theorem 8 for full statement and additional results.*

Interestingly, we are not aware of any classical analogs of such tradeoffs between maximum payment and communication complexity. In particular, the exponential lower bounds against classical auction protocols in [70] hold even with arbitrarily large payments.

[2] Our protocol assumes that we're given some finite upper bound U on valuations, but the communication complexity does not depend on U.

1.1.2 Worst-Case Quantum CC with Two Items

Our second set of results focuses on the particularly simple case of a buyer with additive valuations over only two items, and the Bayesian prior for these values is independent. In this case, classical protocols of [70] already achieve $O(1)$ expected communication, but their worst-case communication is infinite. Note that this is for exactly optimal mechanisms – for approximately optimal mechanisms, worst-case can be reduced to expected communication.

We show that on one hand, a *single qubit* can sometimes replace an *infinite* stream of classical bits.

▶ **Theorem 3** (Separating one-way quantum vs classical)**.** *For the problem of auctioning two items to a single buyer, there is a Bayesian prior over independent item values, such that there is a revenue-optimal and IC one-way quantum auction protocol where the buyer sends 1 qubit and 2 classical bits; yet no finite classical auction protocol can achieve the optimal revenue.*

Furthermore, interactive quantum protocols are even more powerful:

▶ **Theorem 4** (Separating interactive quantum vs one-way quantum)**.** *For the problem of auctioning two items to a single buyer, there is a Bayesian prior over independent item values, such that there is an IC revenue-optimal quantum auction protocol where the seller sends 1 qubit to the buyer, who replies with 1 qubit and 2 classical bits; yet no finite classical or one-way quantum auction protocol can achieve the optimal revenue.*

However, in the worst case, even fully interactive quantum auction protocols cannot achieve optimal revenue in finite worst-case communication.

▶ **Theorem 5** (Limitations of finite interactive quantum auction protocols)**.** *For the problem of auctioning two items to a single buyer, there is a Bayesian prior over independent item values, such that there is a revenue-optimal classical auction protocol that requires a constant number of bits* in exepctation*; yet no worst-case finite IC quantum auction protocol can achieve the optimal revenue.*

1.2 Key Takeaway: Thinking About Quantum and Incentives Together

Communication complexity in game-theoretic setting and quantum communication complexity have each been studied extensively over the past few decades, but in separate lines of work, and to a large extent in disjoint communities. The key takeaway from our work is that we can unlock significant advantages by thinking of both together - advantages that are not possible by composing disjoint results for classical-strategic communication and quantum-cooperative communication.

In particular, it is important to note that our quantum speed-ups are not achievable by a generic quantum speed-up on communication (for example, Holevo's theorem states that an n-qubit quantum state, even with infinite precision, can only carry up to n classical bits accessible information [49]). In fact, they cannot be derived from a quantum speed-up on any non-strategic communication problem.

We further note that our infinite separations require quantum operations with infinite precision. While this is clearly not practical, it is the standard textbook model of quantum computing, and, to the best of our knowledge, no previous work on quantum-cooperative communication complexity exhibits infinite separations[3].

[3] Compared with quantum advantages in interactive proofs, e.g. the celebrated $\mathsf{MIP}^* = \mathsf{RE}$ [51], it is

Surprises for the quantum side:

- **Exponential quantum speedup on a natural problem (Theorem 1).** For combinatorial and unit-demand auctions, we find quantum protocols that are exponentially more efficient than any classical protocol. Existing exponential quantum-classical separations (in cooperative communication complexity) are for problems like Hidden Matching Problem [10] that were designed for the purpose of exhibiting a separation. In contrast, the strategic communication problem we study here was considered before in classical algorithmic game theory [5, 70].
- **Infinite 1-way vs. 1-way + entanglement separation (Theorem 4).** Specifically, we show that a one-way protocol with pre-shared entanglement (an EPR pair) is infinitely more efficient than any one-way protocol with no shared entanglement. This unique separation does not exist in the cooperative quantum communication environment as the honest sender can always prepare the EPR pair and send one half through the channel.
- **Infinite quantum-classical separation (Theorem 3).** We construct an example where we can implement the optimal auction with a one-way quantum communication protocol with 3 qubits in the worst case. However, no worst-case finite classical protocol can implement it.

Figure 1 Summary of most surprising aspects of our results from quantum perspective.

We highlight some of the ways in which our results are distinct from previous work in either line of work in Figure 1 and Figure 2.

Core conceptual idea: Efficient, samplable and verifiable distribution encodings

At a high level, the communication protocol of Bayesian auction boils down to the following task: The seller has a set $\mathcal{S}$ consisting of valid distributions of auction outcomes (allocation and payment). The buyer then selects a distribution D from $\mathcal{S}$ that maximizes his utility. Subsequently, the seller draws a sample from the chosen distribution D and outputs it as the outcome. In general, the complexity of describing a distribution is exponentially higher than specifying an outcome. It is worth noting that this task is completely trivial in a fully cooperative environment as the buyer can simply draw the outcome himself and send only this outcome to the seller. A natural quantum solution to this problem is that the buyer can efficiently encode D in state $\sum_x \sqrt{D_x}\,|x\rangle$. To draw an element from the distribution, the seller only needs to measure this quantum state in the computational basis. However, this simple approach has a caveat – in general it is information theoretically impossible for the seller to verify that this quantum state indeed encodes a valid distribution in set $\mathcal{S}$. To overcome this issue, our paper proposes two solutions. The first one is a general spot-check method – with a small probability, after receiving the quantum state, the seller asks the buyer to send the whole classical description of the distribution; the seller verifies that the classical description is valid and the quantum state is close enough to this classical description. By imposing an exponentially large penalty, we ensure that the buyer is always

worth noting that the latter is only an "unbounded" separation, i.e. that separation still needs the complexity of the MIP^* provers to go to infinity. In contrast, we show separation of 3 qubits vs infinitely many classical bits.

Surprises for the AGT side:

- **Infinite 1-way vs. interactive separation (Theorem 4).** We show an example where an interactive two-way quantum protocol is infinitely better than any one-way quantum protocol. This unique separation does not exist in the classical setting. Since with a trusted party (the seller in our setting), any classical protocol can be "flattened" to a one-way protocol incurring an exponential overhead in the worst-case communication complexity, yet this overhead remains finite.
- **Exponential lower bound on CC $\times$ payment (Theorem 8).** We prove that our exponentially more efficient quantum protocol in Theorem 1 necessitates an exponentially large payment, by establishing an exponential lower bound on the product of payment and communication complexity. This characteristic is distinctive to the quantum setting, as the exponential lower bounds for classical protocols [70] apply even with arbitrarily large payments.

Figure 2 Summary of most surprising aspects of our results from AGT perspective.

incentivized to prepare a quantum state corresponding to a valid distribution while keeping the *expected* communication complexity low. The other solution works for the worst case communication complexity under some specific assumptions on the set of valid distribution of auction outcomes. These assumptions are satisfied for example by the canonical classically-hard example of [24], but not in general (see Theorem 5). By carefully tailoring the quantum protocol to the desired auction, we can ensure that the space of distributions that the buyer can encode coincides with the desired valid space.

1.3 Additional Related Works

Our work extends a rich tradition of studying mechanism design and game theory under the lens of communication complexity – including auctions [61, 12, 4, 31, 28, 26, 27, 14, 2, 15, 33, 3, 5, 76], and also stable matching [41], voting rules [22, 64, 17, 71], fair division [13, 63], computation of equilibria [21, 46, 68, 43, 35, 7, 8, 36, 9], and interdomain routing [53]. In particular, the communication complexity of IC implementing a mechanism vs that of (non-IC) computing the outcome was the focus of [34, 29, 69, 30].

We show exponential separations (and for worst-case complexity – infinite separations) between quantum and classical communication complexity of auctions. Earlier works on separating the two measures (in non-strategic settings) include general boolean functions (constructed for obtaining separations) [65, 10, 37, 38, 66], sampling [1, 57], and very recently also linear regression [58, 72].

Our work is also related to works on quantum game theory – including nonlocal games [19, 20], quantization of classical games [32, 56], quantum equilibria [25, 75], and quantum interactive games [45]. In particular, quantum interactive strategies are also studied in quantum interactive proofs [74, 11, 50, 59, 51].

1.4 Organization

We begin with a review of the quantum communication model and mechanism design in Section 2 and 3, and then introduce our main model of quantum auction protocols in Section 4. Part I brings our results for expected communication: Our quantum auction protocol for

combinatorial valuations in Section 5, and our lower bound against quantum auction protocols with bounded maximum payment in Section 6. Part II focuses on worst-case communication: we begin with preliminaries of optimal 2-item auctions in Section 7; in Section 9 we construct an example separating one-way quantum auction protocols from finite classical; in Sections 8 and 10 we separate interactive quantum auction protocols from one-way; and finally in Section 11 we show that in general no finite quantum auction protocol can guarantee optimal revenue.

2 Preliminaries I: Quantum

Bra-ket notation

In this paper, we may occasionally use bra-ket notation. Specifically, within an N-dimensional complex vector space, we represent each unit-length column vector as a ket, denoted as $|\phi\rangle$. Correspondingly, for every unit-length vector $|\phi\rangle$, a bra $\langle\phi|$ is defined as an N-dimensional row vector that is the conjugate transpose of $|\phi\rangle$.

Moreover, we use the notation $|a\rangle$ for $a \in \{1, \ldots, N\}$ to indicate the column vector with a value of 1 at the a-th coordinate and 0 in all other positions. We refer to $|1\rangle, \ldots, |N\rangle$ as the computational basis.

We employ the notation $|\phi\rangle$ to represent a pure quantum state associated with the density matrix $|\phi\rangle\langle\phi|$. Inversely, a quantum state described by the density matrix ρ is considered a pure state if there exists a $|\phi\rangle \in \mathbb{C}^N$ such that $\rho = |\phi\rangle\langle\phi|$.

Closeness of states

Given two positive semidefinite matrices $\rho, \sigma \in \mathbb{C}^{N\times N}$, the trace distance between them is defined as

$$T(\rho,\sigma) = \max_F \frac{1}{2}\sum_\ell \left|\mathrm{Tr}(F_\ell\rho) - \mathrm{Tr}(F_\ell\sigma)\right|,$$

where $\{F_\ell\}$ is maximized over all possible POVMs[4].

In particular, when ρ and σ are density matrices, $T(\rho,\sigma)$ is equal to the total variation distance between classical distributions obtained by measuring two states maximized over all possible measurements.

Below is an important property of the trace distance:

$$T(\rho,\sigma) \le \sqrt{1-\mathrm{Tr}(\rho\sigma)}. \tag{1}$$

2.1 (Non-Strategic) Quantum Communicatiom Protocols

We give an overview of multi-party quantum communication protocols. For readers who are familiar with quantum communication, this model is equivalent to the ones used in the literature (e.g. [77]). A formal description of two-party strategic communication model is given in Section 4. A multi-party quantum communication protocol is defined over a system of qubits, that are initially partitioned between the parties. The protocol proceeds in rounds; in each round, one party can locally manipulate or measure her qubits, and then send a

[4] a positive operator-valued measure (POVM) is a finite set of positive semidefinite Hermitian matrices that sum to identity.

subset of them to other parties. The *communication complexity* of a protocol is the total number of qubits sent by parties across all rounds. We now provide more detail on each of those components.

Quantum systems

Let m be an upper bound on the number of qubits in the system.[5] The state $\rho^{(r)}$ of the system at the beginning of round r can be mathematically represented as a *density matrix*[6] $\rho^{(r)} \in (\mathbb{C}^{2\times 2})^{\otimes m}$. Note that $(\mathbb{C}^{2\times 2})^{\otimes m} = \mathbb{C}^{2^m \times 2^m}$, i.e. it is just a 2^m-by-2^m complex matrix; however, the former tensor product notation will be useful when we consider the qubits held by each party.

Initial state of the system

Initially, each party holds $m_i^{(0)}$ qubits ($\sum_i m_i^{(0)} = m$). Because we're concerned with quantum protocols for mechanisms with classical inputs, we assume that initially all the qubits are not entangled (e.g. the initial state is $\rho^{(0)} = (|0\rangle\langle 0|)^{\otimes m}$). In particular, it is important that qubits held by different parties are initially non-entangled.

Local histories

A party's local history consists of the number of qubits that she sent and received in each round so far in the protocol, as well as the outcomes of measurements that she locally performed on her qubits (see more on measurements below).

Local manipulations: unitary operators and measurements

In each round, before sending any qubits, the active party can locally apply quantum operations and measurements to the qubits that she currently holds. If $m_i^{(r)}$ is the number of qubits held by party i at the beginning of round r, we can represent the state $\rho^{(r)}$ as a density matrix in $(\mathbb{C}^{2\times 2})^{\otimes m_i^{(r)}} \otimes (\mathbb{C}^{2\times 2})^{\otimes m - m_i^{(r)}}$. Party i's operations can transform the state into

$$\rho^{(r+1/2)} = (U \otimes I_{2^{m-m_i^{(r)}}})^\dagger \rho^{(r)} (U \otimes I_{2^{m-m_i^{(r)}}}),$$

where $I_{2^{m-m_i^{(r)}}}$ is the identity operator on qubits held by other parties, U is a unitary operator of i's choice, acting only on i's qubits, and $\rho^{(r+1/2)}$ is the new state of the quantum system after applying the operator (but before the measurement).

Similarly, party i can measure her qubits. A POVM is defined by L matrices $\{A_\ell\}_{\ell=1}^L \in \mathbb{C}^{2^{m_i^{(r)}} \times 2^{m_i^{(r)}}}$ such that $\sum_\ell A_\ell^\dagger A_\ell = I_{2^{m_i^{(r)}}}$. After applying the measurement, with probability

$$\mathrm{Tr}\left(\left(A_\ell \otimes I_{2^{m-m_i^{(r)}}}\right)^\dagger \left(A_\ell \otimes I_{2^{m-m_i^{(r)}}}\right) \rho^{(r+1/2)}\right),$$

[5] We assume for simplicity of notation that there is a finite upper bound on the total number of qubits. Our results can be generalized e.g. to a setting where each party can add qubits in each round of the protocol, and a setting where local operators are defined by general quantum channels.

[6] A matrix is a *density matrix* if it is a positive semidefinite, trace 1 Hermitian matrix. *Hermitian* means that $A^\dagger = A$, where $A^\dagger$ is the conjugate transpose of A.

the state of the system is updated to

$$\rho^{(r+1)} = \frac{\left(A_\ell \otimes I_{2^{m-m_i^{(r)}}}\right)^\dagger \rho^{(r+1/2)} \left(A_\ell \otimes I_{2^{m-m_i^{(r)}}}\right)}{\mathrm{Tr}\left(\left(A_\ell \otimes I_{2^{m-m_i^{(r)}}}\right)^\dagger \left(A_\ell \otimes I_{2^{m-m_i^{(r)}}}\right) \rho^{(r+1/2)}\right)}.$$

It is wlog for each party to perform the measurement after all the unitary operators in a given round.

Sending and receiving qubits

After applying local operations, the active party sends exactly one of the qubits she holds to another party. Sending qubits does not change the state of the quantum system, but it changes which party can operate on the sent qubits.

Note that it is wlog to send e.g. the last qubit, because locally qubits can be swapped by unitary operators.

Termination of the protocol

The protocol may terminate after a pre-determined number of rounds, or by any party as a function of her private inputs and/or local history.

Complexity of the protocol

The main metric of complexity of the protocol that we use is the total number of qubits sent by different parties. We give bounds for the complexity of both in-expectation (over the outcome of quantum measurements) and worst-case communication. In addition to the total number of qubits, we will show that: (i) In some cases it is possible to simplify protocols by replacing some qubits with classical bits, and (ii) we also consider the effect of restricting the number of rounds of the protocol.

2.1.1 Conventions

We make the following conventions to simplify both our notation and analysis:

- There is no seperate channel for classical information. For convenience, when we say a message is intended to be classical, it means the receiver immediately measures the qubit in the computational basis $(|0\rangle, |1\rangle)$.
- For convenience, in a 2-party protocol, when we mention that the a player sends m consecutive qubits to the other, it technically means that this player sends these qubit over m rounds and the other player responds with a dummy qubit in each round.

2.1.2 Choi-Jamiołkowski representation of protocols and strategies

Consider a quantum protocol with a fixed number of rounds R and a fixed number of qubits sent in each round. A *strategy* s_i of a party i who is active in R_i rounds is a sequence of R_i mappings applied to the qubits that it holds at each round, together with a measurement of its qubits at the end of the protocol. A *co-strategy* s_{-i} is a sequence of mappings by other parties at their rounds (and finally a measurement). Notice that the tuple of protocol, strategy, and co-strategy, fully determine the distribution of measurement realizations at the end of the protocol.

Suppose that party i's measurements have L_i possible outcomes and the other parties have L_{-i} possible outcomes. [45] show that any strategy (resp, co-strategy) can be represented as L_i (resp. L_{-i}) matrices of dimension that depend only on the communication complexity, and not on the additional (possibly very large) quantum memory of the parties. For ease of presentation, we state only the simplest form of the theorem that we need; in particular, we avoid the notation necessary for actually defining the Choi-Jamiołkowski representation.

▶ **Theorem 6** (Choi-Jamiołkowski representation of strategies in interactive quantum protocols [45])**.** *Consider any R-round quantum protocol with communication complexity K, a party i in the protocol, and strategy s_i for i and co-strategy s_{-i} for the rest of the parties. Let $\Phi^{(s_i)}, \Psi^{(s_{-i})}$ denote the respective Choi-Jamiołkowski representation. Then the probability of measuring outcome (a_i, a_{-i}) is given by*

$$2^K \cdot \mathrm{Tr}\left(\Phi^{(s_i)}_{a_i} \Psi^{(s_{-i})}_{a_{-i}}\right).$$

Moreover, each of $\Phi^{(s_i)}_{a_i}$, $\Psi^{(s_{-i})}_{a_{-i}}$ is a 2^{2K} by 2^{2K} positive semidefinite Hermitian matrix, and $\sum_{a_i=1}^{L_i} \mathrm{Tr}\left(\Phi^{(s_i)}_{a_i}\right) = \sum_{a_{-i}=1}^{L_{-i}} \mathrm{Tr}\left(\Phi^{(s_{-i})}_{a_{-i}}\right) = 1$[7]*.*

3 Preliminaries II: Mechanism Design

We consider the mechanism (auction) design for selling n indivisible items to a single risk-neutral buyer. A buyer has a *type* (valuation function) $v : 2^{[n]} \to \mathbb{R}_{\geq 0}$ specifying his value for each bundle (subset). We use X to denote the type set, which contains all possible types of the buyer. For our purposes, it is important to define the two simplest and most widely studied class valuations:

- **Additive** If there exists a value of each item $v_1, \ldots, v_n$ such that $v(S) = \sum_{i \in S} v_i$.
- **Unit-demand** If there exists a value of each item $v_1, \ldots, v_n$ such that $v(S) = \max_{i \in S} v_i$.

Some of our results also hold for more general classes of valuations[8], which satisfy the following hierarchy of increasing generality:

$$\text{additive, unit-demand} \subset \text{gross-substitutes} \subset \text{submodular} \subset \text{XOS} \subset \text{subadditive} \subset \text{combinatorial}.$$

In addition to satisfying these structures, valuations are usually assumed to be monotone; our results hold for both monotone and non-monotone valuations.

Without loss of generality, we consider direct mechanisms. The buyer reports a type $v' \in X$ to the mechanism, and the mechanism then allocates a (randomized) bundle to the buyer and charges the buyer a price. $\mathcal{M}$ consists of two functions.

- An allocation function $\mathcal{A} : X \to [0,1]^{2^{[n]}}$ gives the probability of allocating each bundle to the buyer declares to have each possible type.
- A payment function $\mathcal{Q} : X \to \mathbb{R}_{\geq 0}$ gives the price the buyer needs to pay for each declared type of the buyer.

Let D be a distribution over bundles. With a slight abuse of notation, we denote by $v(D) = \mathbb{E}_{S \sim D} v(S)$ the expected value of the buyer with type v.

---

[7] In the original definition of [45], Choi-Jamiołkowski representations $\Phi^{(s_i)}, \Psi^{(s_{-i})}$ are not normalized. Here, we normalize all Choi-Jamiołkowski representations (now they are all density matrices), that is why we have an additional 2^K factor in the probability of outcome compared to the original paper.

[8] See e.g. [52] for definitions; they are not necessary for understanding our paper.

We say a mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{Q})$ is *incentive compatible* (IC) if

$v(\mathcal{A}(v)) - \mathcal{Q}(v) \geq v(\mathcal{A}(v')) - \mathcal{Q}(v') \quad \forall v, v' \in X.$

We say a mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{Q})$ is ε-*incentive compatible* (ε-IC) if

$v(\mathcal{A}(v)) - \mathcal{Q}(v) \geq v(\mathcal{A}(v')) - \mathcal{Q}(v') - \varepsilon \quad \forall v, v' \in X.$

We say a mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{Q})$ is *individually rational* (IR) if

$v(\mathcal{A}(v)) - \mathcal{Q}(v) \geq 0 \quad \forall v \in X.$

We say a mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{Q})$ is ε-*individually rational* (ε-IR) if

$v(\mathcal{A}(v)) - \mathcal{Q}(v) \geq -\varepsilon \quad \forall v \in X.$

For a mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{Q})$, we denote by $u : X \to \mathbb{R}$ the expected utility of the buyer when he truthfully reports the valuation function. It follows from the definition that $u(v) = v(\mathcal{A}(v)) - \mathcal{Q}(v)$.

Revenue Maximization

In this paper, we primarily focus on the revenue-optimal Bayesian mechanism design. In this setting, the buyer knows his type v for sure. However, the seller only knows the probability distribution over X. Let $f : X \to \mathbb{R}$ be the probability density function of this distribution.

The goal of revenue-optimal Bayesian mechanism design is to find an IC and IR mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{Q})$ that maximizes the revenue of the seller:

$$Rev = \int_X \mathcal{Q}(v) f(v) \mathrm{d}v.$$

4 Quantum Communication Model with a Strategic Player

We now introduce the main strategic communication model of this work, which formally defines the elements of a two-player quantum communication protocol subject to strategic manipulation. In essence, when the length of the protocol is fixed, this model is equivalent to the one used in the literature of quantum games and quantum interactive proofs (see e.g. [45]). The communication is between one strategic player (we call it the buyer), and a truthful player (we call it the seller). In this setup, the seller initially possesses n qubits, while the buyer has m qubits, with S representing the finite set of possible communication outcomes. Initially, the joint state is $|0\rangle^{\otimes(n+m)}$. The communication proceeds in rounds (or steps). Since we will cover protocols with infinite worst-case communication complexity (but bounded expected communication), we do not specify the total number of rounds in our model. In each round, one of the player performs a local operation on qubits in their hand and then sends *one* qubit to the other player. Or you can alternatively think there is a one-qubit register shared by both players. We assume the seller takes the first round and then they alternate in the following rounds.

The seller's operations

Without loss of generality, we assume the seller only performs general measurements in her rounds[9]. For round i, let $\{A_x^i\}_{x \in S \cup \{\perp\}}$, such that $\sum_{x \in S \cup \{\perp\}} (A_x^i)^\dagger A_x^i = I_{2^n}$, be the seller's

[9] If the seller simply want to apply a unitary U, she can do it by letting $A_\perp^i = U$, and $A_x^i = 0$ for any $x \in S$.

measurements. Let $h_i \in S \cup \{\perp\}$ be the measurement outcome of round i. For each round i, if $h_i = \perp$ then communication continues, otherwise the seller terminates the communication and outputs h_i as the outcome[10].

The buyer's operations

In our model, only seller can terminate the protocol and output an outcome. Therefore, by the principle of deferred measurement (see e.g. Chapter 4.4 of [60]), we wlog assume the buyer makes no measurement[11], and his only local operation in round i is a unitary U_i with dimension 2^{m+1}, for $i = 2, 4, \ldots$.

The seller's strategy

A seller's strategy includes the following elements:

- Set of communication outcomes: S.
- The size of initial local quantum memory: n.
- General measurements: $\{A_x^i\}_{x \in S \cup \{\perp\}}$, for $i = 1, 3, 5, \ldots$.

The buyer's strategy

A buyer's strategy s^{buy} includes the following elements:

- The size of initial local quantum memory: m.
- Unitary operators: U_i, for $i = 2, 4, 6, \ldots$.

Quantum auction protocols

Our objective is to implement an auction using the strategic quantum communication model previously outlined. Our quantum auction protocols are a generalization of classical auction protocols defined in [70]. The classical auction protocols are also a special case of Bayesian incentive compatibility (BIC)-incentivizable binary dynamic mechanism (BDM) defined in [34]. For simplicity, here we only define the quantum analog for auctions, but exploring the quantum communication complexity of mechanisms more broadly is an interesting direction for future work.

▶ **Definition 7** (Quantum auction protocols). *A* quantum auction protocol $\mathcal{P}$ *that sells* n *items to a single buyer with type space* X *consists of:*

- *A seller's strategy* $s^{\mathcal{P}}_{seller}$ *such that the outcomes in the outcome set* S *are in the form* (B, p)*, where* $B \subseteq [n]$ *is a subset of items and* $p \in \mathbb{R}_{\geq 0}$ *is the price.*
- *A suggested strategy function* $s^{\mathcal{P}}_*$ *that maps each valuation in the type space* X *to a buyer's strategy.*

[10] By the principle of deferred measurement (see e.g. Chapter 4.4 of [60]), any non-terminating measurements outcomes can be removed by adding more qubits to the system. Therefore, it is wlog to only consider measurement with at most one non-terminiating outcome($\perp$) each round.

[11] By the principle of deferred measurement, the buyer can always obtain the same outcome distribution by adding more qubits to his local memory and removing measurements. Since the buyer is allowed to choose an arbitrarily large memory size m (we will discuss it later), it is wlog not to consider buyer's measurement.

Given a seller's strategy s_{seller}, and a buyer's strategy s_{buyer} let $D(s_{\text{seller}}, s_{\text{buyer}})$ be the outcome distribution of the communication[12]. Let $\mathcal{A}(s_{\text{seller}}, s_{\text{buyer}})$ be the marginal distribution of the first component of $D(s_{\text{seller}}, s_{\text{buyer}})$. So, $\mathcal{A}(s_{\text{seller}}, s_{\text{buyer}})$ is a distribution over subsets of $[n]$. Let $\mathcal{Q}(s_{\text{seller}}, s_{\text{buyer}})$ be the expected value of the second component (price) of $D(s_{\text{seller}}, s_{\text{buyer}})$.

With definitions above, we say a quantum auction protocol $\mathcal{P}$ *implements* the mechanism

$$\mathcal{M}^{\mathcal{P}} = \left(\mathcal{A}(s^{\mathcal{P}}_{\text{seller}}, s^{\mathcal{P}}_{*}(\cdot)), \mathcal{Q}((s^{\mathcal{P}}_{\text{seller}}, s^{\mathcal{P}}_{*}(\cdot)))\right).$$

Further, we say a quantum auction protocol $\mathcal{P}$ is ε-IC if for any type $v \in X$, and any buyer's strategy $\hat{s}$, the following holds,

$$v\left(\mathcal{A}(s^{\mathcal{P}}_{\text{seller}}, s^{\mathcal{P}}_{*}(v))\right) - \mathcal{Q}(s^{\mathcal{P}}_{\text{seller}}, s^{\mathcal{P}}_{*}(v)) \geq v\left(\mathcal{A}(s^{\mathcal{P}}_{\text{seller}}, \hat{s})\right) - \mathcal{Q}(s^{\mathcal{P}}_{\text{seller}}, \hat{s}) - \varepsilon.$$

By definition, the mechanism implemented by an ε-IC quantum auction protocol is an ε-IC mechanism.

We say quantum auction protocol $\mathcal{P}$ is ε-IR if for any type $v \in X$ the following holds,

$$v\left(\mathcal{A}(s^{\mathcal{P}}_{\text{seller}}, s^{\mathcal{P}}_{*}(v))\right) - \mathcal{Q}(s^{\mathcal{P}}_{\text{seller}}, s^{\mathcal{P}}_{*}(v)) \geq -\varepsilon.$$

By definition, the mechanism implemented by an ε-IR quantum auction protocol is an ε-IR mechanism.

Exactly IC and IR quantum protocols are defined similarly.

The main goal of our paper is to find an (ε-)IC and (ε-)IR quantum auction protocol that implements the revenue-optimal mechanism.

Part I
Expected communication

5 ε-IC Quantum Protocols for General Valuations

Consider selling n items to a single buyer with combinatorial valuations drawn from prior $\mathcal{D}$. We will show that for any direct IC mechanism $\mathcal{M}$ that only ever allocates B different deterministic bundles, there is an ε-IC quantum protocol with the same expected payment for every type using $O(\log B)$ qubits of communication in expectation. Note that our protocol holds for arbitrarily small ε, and the constant factor in $O(\log B)$ does not depend on ε.

Moreover, by employing a standard ε-IC-to-IC reduction that discounts all the payments by $(1-\sqrt{\varepsilon})$-factor, we can transform an ε-IC quantum protocol into an exactly IC quantum protocol that has the same expected communication complexity, while incurring only a negligible loss in revenue (see e.g. [42, Theorem 7]).

▶ **Theorem** (Theorem 1 restated). *Let $\mathcal{D}$ be a prior over buyer's combinatorial valuations over n; assume all valuations are in the range $[0, U]$. Let $\mathcal{M}$ be any mechanism that can only possibly allocate one of B subsets of the items. Finally, let $\delta > 0$ be any parameter (δ may be a function of n or B). Then there is an IC auction quantum protocol that guarantees a $(1-\delta)$-fraction of $\mathcal{M}$'s expected revenue using $O(\log(B))$ qubits in expectation.*

[12] Throughout the paper we only consider seller's strategies which guarantee termination within finite steps with probability 1 for any buyer's strategy.

We first present a modification of an $O(B\log(B))$-bits classical protocol given in [70]. Then, we augment it with a single quantum message from the buyer to the seller that gives an exponential improvement in the expected communication.

5.1 Warm-up: An Inefficient Classical Protocol [70]

First note that if the maximum value of the buyer over any bundle is at most U, any IC and IR direct mechanism $\mathcal{M}$ that only ever allocates B different bundles can be converted into an equivalent direct mechanism $\mathcal{M}'$ that only ever allocates B different bundles and always charges either 0 or U. This implies the new mechanism $\mathcal{M}'$ has $2B$ different deterministic outcomes (each deterministic outcome is a bundle-payment pair). Suppose B different bundles allocated in $\mathcal{M}$ are $\pi_1, \ldots, \pi_B$, then $\mathcal{M}'$ creates 2 outcomes $(\pi_j, 0), (\pi_j, U)$ for each bundle in $\mathcal{M}$. For each type, v, $\mathcal{M}$ has a distribution over B bundles and an expected payment P (by IR, $P \leq U$). Then $\mathcal{M}'$ defines a distribution over $2B$ outcomes for each type v: suppose in $\mathcal{M}$ the proability of receiving π_j is p_j, $\mathcal{M}'$ gives outcome (π_j, U) with probability $p_j \cdot \frac{P}{U}$ and outcome $(\pi_j, 0)$ with probability $p_j \cdot (1 - \frac{P}{U})$.

We first present a classical (randomized) IC protocol that implements $\mathcal{M}'$ with $O(B \log B)$ expected communication complexity. This protocol differs slightly from the one in [70] and serves as a more straightforward foundation for constructing a quantum protocol. A probability distribution over $2B$ outcomes can be represented by $2B$ non-negative real numbers $p_1, \ldots, p_{2B}$ such that $\sum_{i=1}^{2B} p_j = 1$. We call a distribution *feasible* if it is a distribution over $2B$ outcomes for some type in mechanism $\mathcal{M}'$. At each round, the buyer sends $2B$ bits, and the seller either terminates the protocol with an allocation and a payment or continues by moving to the next round and letting the buyer send more bits.

Buyer's suggested strategy

Given the buyer's type and mechanism $\mathcal{M}'$, we denote by $p_1, \ldots, p_{2B}$ the probability of each outcome. The buyer sends $2B$ bits each round. The suggested strategy is to send, in the r-th Buyer round, the r-th bit of the binary representation of $p_1, \ldots, p_{2B}$, respectively.

Seller's strategy

After receiving each bit, the seller first checks if all buyer's messages so far are consistent with some feasible distributions, which means messages are binary prefixes of probabilities corresponding to some feasible distributions. When there is only one possible value for the next bit that is consistent with some feasible distribution, then the seller sets the value of the next bit of the message to be the only feasible value and ignores the original bit of the message.

Let $m_j^{(r)}$ be the j-th bit of message receives at round r. For round r, we denote by $\tau^{(r)}$ the total probability revealed so far.

In addition, we define $\tau^{(0)} = 0$. After receiving all $2B$ bits of message at round r, the protocol terminates with probility $\frac{\tau^{(r)} - \tau^{(r-1)}}{1 - \tau^{(r-1)}}$. Conditioned on terminating, the protocol assigns allocation and payment of outcome j of $\mathcal{M}'$ with probability $\frac{m_j^{(r)} \cdot 2^{-r}}{\tau^{(r)} - \tau^{(r-1)}}$ for each j. Finally, with probability $1 - \frac{\tau^{(r)} - \tau^{(r-1)}}{1 - \tau^{(r-1)}}$, the protocol continues. It is important to note that the protocol terminates in no longer than r rounds with probability $\tau^{(r)}$.

IC

The suggested strategy is optimal for the buyer.

Communication complexity

The overall expected communication complexity is $O(B \log B)$.

5.2 Quantum Protocol

The idea of our quantum protocol is to replace all classical bits the buyer sends in the first $2 \log B$ rounds (total of $O(B \log B)$ bits) with a single message with $\log(B)+1$ qubits and $2 \log B$ classical bits. Note that $\log(B)+1$ qubits can encode an arbitrary distribution with support size $2B$ (see details below). The buyer is supposed to encode the distribution associated with his type into a quantum state. Then the seller can measure the quantum state and decide the allocation and payment according to the outcome. Here, the problem with this approach is that the buyer might encode an infeasible distribution (not associated with any type), and the seller has no way to tell if a quantum state encodes a feasible distribution.

To overcome this challenge, the seller will, with high probability, blindly trust the buyer and determine the allocation and payment according to the measurement outcome. However, with a small probability, rather than measuring the qubits, the seller asks the buyer to reveal the probability distribution he encoded by sending its full description classically. Subsequently, the seller can perform a measurement to verify if the quantum state accurately encodes the given distribution. Specifically, any quantum state that unfaithfully encodes the distribution will fail the test with a non-zero probability. As a result, the protocol can penalize the buyer with a big payment once she observes that the test has failed.

Seller's strategy

In the first round, the seller expects a quantum message of $(\log(B)+1)$ qubits, which we denote by m_Q, and a $\log(B+1)$-bit classical message that represents an integer $\hat{\tau} \in [0, B]$.

With probability $\gamma(\hat{\tau}) = 1 - \frac{1}{2B} - (1 - \frac{1}{2B})\frac{\hat{\tau}}{B^2}$, the seller measures the quantum message m_Q in the computational basis $(|1\rangle, |2\rangle, \ldots, |2B\rangle)$ and terminates the protocol. Suppose the measurement outcome is $|a\rangle \in \{|1\rangle, |2\rangle, \ldots, |2B\rangle\}$, the allocation and payment are determined according to outcome a of mechanism $\mathcal{M}'$.

With probability $1 - \gamma(\hat{\tau}) = \frac{1}{2B} + (1 - \frac{1}{2B})\frac{\hat{\tau}}{B^2}$, the seller asks the buyer to send $4B \log B$ classical bits to represent $2B$ binary numbers, $\widehat{p_1}, \ldots, \widehat{p_{2B}}$, each consisting of $2 \log B$ bits (we denote by m_C this classical message). Applying the same correction procedure as described in Subsection 5.1, we can assume that $\widehat{p_1}, \ldots, \widehat{p_{2B}}$ is the rounding-down to nearest multiple of $1/B^2$ of a feasible distribution $p_1, \ldots, p_{2B}$. The seller then verifies that $\hat{\tau} = B^2 \cdot (1 - \sum_i \widehat{p_i})$; if it isn't, the protocol terminates with the empty allocation and payment $2BU^3\varepsilon^{-2}$. Next, the seller measures the quantum message she receives earlier m_Q in a way such that the measurement has two outcomes $0, 1$, and the probability of outcome 1 is given by $\mathrm{Tr}(\rho |\psi\rangle \langle\psi|)$, where ρ is the reduced density matrix that represents the state of m_Q at the time of measurement, and $|\psi\rangle$ is the canonical state of classical message m_C that is given by $|\psi\rangle = \frac{B}{\sqrt{B^2 - \hat{\tau}}} \sum_{i=1}^{2B} \sqrt{\widehat{p_i}}\, |i\rangle$.

If the measurement outcome is 0, the protocol terminates with empty allocation and payment $2BU^3\varepsilon^{-2}$. Otherwise, the protocol continues as a purely classical protocol in the following manner: for each $i \in \{1, \ldots, 2B\}$, with probability $\frac{1}{2B(1-\gamma(\hat{\tau}))}\widehat{p_i}$, the protocol terminates with outcome i of mechanism $\mathcal{M}'$. Finally, with probability $1 - \frac{1}{2B(1-\gamma(\hat{\tau}))}\sum_i \widehat{p_i}$, the seller starts to run the classical protocol from round $2 \log B + 1$ and let $\widehat{p_1}, \ldots, \widehat{p_{2B}}$ be the message she received in the first $2 \log B$ rounds.

Buyer's suggested strategy

Given the buyer's type and mechanism $\mathcal{M}'$, we denote by $p_1, \ldots, p_{2B}$ the probability of each outcome. Let $\widetilde{p_1}, \ldots, \widetilde{p_{2B}}$ be the numbers such that for any i, the binary representation of $\widetilde{p_i}$ consists of first $2 \log B$ bits of the binary representation of p_i. The suggested strategy is to send, for quantum message m_Q whose density matrix is given by $|\varphi\rangle\langle\varphi|$, where $|\varphi\rangle$ is defined as $|\varphi\rangle = \frac{B}{\sqrt{B^2-\tilde{\tau}}} \sum_{i=1}^{2B} \sqrt{\widetilde{p_i}}\,|i\rangle$, and send integer $\tilde{\tau} = B^2(1 - \sum_i \widetilde{p_i})$. In addition, when the protocol asks the buyer to send classical message m_C, the suggested strategy sends $\widetilde{p_1}, \ldots, \widetilde{p_{2B}}$. Finally, as for the classical protocol part, for round corresponding to the r-th round of the classical protocol, the suggested strategy is to send the r-th bit of the binary representation of $p_1, \ldots, p_{2B}$, respectively.

ε-IC

We demonstrate that this protocol is ε-IC.

Communication complexity

The overall expected communication complexity is $O(\log B)$.

6 Lower Bounds for Quantum Protocols with Small Payments

Throughout this section, we normalize the valuations so that the upper bound on the highest value is $U = 1$.

▶ **Theorem 8** (Full version of Theorem 2). *Let n be the number of items, P be the upper bound on the payments and in the quantum auction protocol, and $\hat{K}$ an upper bound on the expected communication complexity. Then we have the following lower bounds on $\hat{K}P$:*

- *For unit-demand valuations, any quantum auction protocol that obtains $\Omega(1)$-approximation to the optimal revenue must satisfy $\hat{K}P = \Omega(n)$.*
- *For Gross-substitutes valuations, any quantum auction protocol that obtains $\Omega(1)$-approximation to the optimal revenue must satisfy $\hat{K}P = 2^{\Omega(n^{1/3})}$.*
- *For XOS valuations, any quantum auction protocol that obtains $\Omega(1)$-approximation to the optimal revenue must satisfy $\hat{K}P = 2^{\Omega(n)}$.*
- *For XOS valuations over independent items, any quantum auction protocol that obtains $4/5 + \Omega(1)$-approximation to the optimal revenue must satisfy $\hat{K}P = 2^{\Omega(n)}$.*

6.1 A List-Decodable Code of Bayesian Priors

The lower bounds of [70] against approximately optimal classical auction protocols construct, for each valuation class (unit-demand, submodular, etc.), a family of Bayesian priors over valuations from this class. Each family has the following "list-decodability" guarantee: no mechanism can simultaneously obtain high revenue on a "list" of $\omega(1)$ priors from the family. We now state the results from [70], with different parameters for family size and approximability for different valuation classes.

▶ **Lemma 9** (Family of hard priors [70]). *For each of the following combinations of valuation class X over n items, family size ζ, and approximability factor γ, there exists a family of ζ Bayesian priors over valuations from X, and a small constant $\varepsilon > 0$ such that no single ε-IC and ε-IR mechanism can simultaneously obtain a γ-approximation of the optimal revenue from $\omega(1)$ distinct priors from the family.*

- X = *unit-demand;* $\zeta = 2^{2^{\Omega(n)}}$; γ = *arbitrarily small constant.*
- X = *gross-substitute;* $\zeta = 2^{2^{2^{o(n^{1/3})}}}$; γ = *arbitrarily small constant.*
- X = *XOS;* $\zeta = 2^{2^{2^{\Omega(n)}}}$; γ = *arbitrarily small constant.*
- X = *XOS over independent items;* $\zeta = 2^{2^{2^{\Omega(n)}}}$; $\gamma = 4/5 + \tau$ *for arbitrarily small constant* τ.

6.2 Approximate Cover over Efficient Quantum Auction Protocols

Our next objective is to demonstrate the existence of a finite set of protocols capable of approximating any (almost) IC and (almost) IR quantum protocol with bounded worst-case communication complexity. Formally speaking, we have the following lemma.

▶ **Lemma 10** (Approximate cover over efficient quantum auction protocols). *Let X be a type space with n items, B an upper bound of number of feasible bundles, P an upper bound of payments and values of types in X, $\varepsilon > 0$ as an approximation parameter, and K an upper bound on the communication complexity, there exists a set $\mathcal{S} = \mathcal{S}(n, B, P, \varepsilon, K)$ of mechanisms such that the following hold:*

- *Small cover:* $|\mathcal{S}| = 2^{2^{O(K+\log B+\log\log P+\log n)}}$.
- *Mechanisms in $\mathcal{S}$ are $2\sqrt{\varepsilon}$-IC and $2\sqrt{\varepsilon}$-IR.*
- *Approximate covering property: For any ε-IC and ε-IR quantum protocol $\mathcal{P}$ with a worst-case communication complexity bounded by K, let $\mathcal{M}_{\mathcal{P}} = (\mathcal{A}_{\mathcal{P}}, \mathcal{Q}_{\mathcal{P}})$ be the mechanism induced by $\mathcal{P}$. Then there exists a mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{Q}) \in \mathcal{S}$ such that for every type $v \in X$, we have*

$$\mathcal{Q}(v) \geq (1 - \sqrt{\varepsilon})\mathcal{Q}_{\mathcal{P}}(v).$$

Part II
Worst-case communication

7 Preliminaries III: Optimal Mehchanisms for Selling 2 Items

In this section, we introduce a special case of framework of [24, 39] to characterize optimal auctions for selling two goods to a single additive buyer with independent valuations. We will first give a picture of their framework and then discuss how to apply duality theory to show that infinite menu complexity (aka infinite worst-case classical communication complexity) is inevitable for the optimal mechanisms of some prior distributions.

For simplicity, we only consider the case where the buyer's type space X is $[0,1]^2$, and each coordinate represents the buyer's value of each good. We assume the prior distribution has a density function $f(x,y) = f_1(x)f_2(y)$. Further, we assume f_1, f_2 are continuous and differentiable with bounded derivatives.

It is well-known (see e.g. [67, 54], that for any IC and IR mechanism the utility function $u : [0,1]^2 \to \mathbb{R}$ is convex, non-negative, non-decreasing, and 1-Lipschitz with respect to the ℓ_1 norm. Also, given any $u : [0,1]^2 \to \mathbb{R}$ with these conditions, the utility function uniquely[13] defines an IC and IR mechanism $\mathcal{M}$ with allocation function $\mathcal{A}(v) = \nabla u(v)$ and payment function $\mathcal{Q}(v) = \mathcal{A}(v) \cdot v - u(v)$.

[13] Up to measure zero.

At a high level, the mechanisms established in [24, 39] partition $[0,1]^2$ into 4 regions $Z, \mathcal{A}, \mathcal{B}, \mathcal{W}$ induced by a convex area Z defined by two concave functions s_1, s_2 and a straight line $x + y = P_{\text{crit}}$ for some $P_{\text{crit}} \in [0,2]$. Moreover, [24] calls $Z, \mathcal{A}, \mathcal{B}, \mathcal{W}$ *the canonical partition with respect to* Z, and P_{crit} *the critical price*. For ease of exposition, we only consider the case that the line $x + y = P_{\text{crit}}$ intersects both curves $x = s_1(y)$ and $y = s_2(x)$.

More precisely, let $s_1, s_2 : [0,1] \to [0,1]$ be two 1-Lipschitz, concave and non-increasing function, and $P_{\text{crit}} \in [0,2]$. Let x_{crit} be the solution to $s_2(x) = P_{\text{crit}} - x$, and y_{crit} be the solution to $s_1(y) = P_{\text{crit}} - y$. We can always find such solutions since we assume the line $x + y = P_{\text{crit}}$ intersects both curves.

Region $Z \subseteq [0,1]^2$ is defined as the region enclosed by s_1, s_2, and line $x + y = P_{\text{crit}}$. Formally, $Z = \{(x,y) \in [0,1]^2 : x \le s_1(y), y \le s_2(x), x + y \le P_{\text{crit}}\}$. The other three regions are defined as follows:

$$\mathcal{A} = \{(x,y) \in [0,1]^2 : x < x_{\text{crit}}\} \backslash Z; \quad \mathcal{B} = \{(x,y) \in [0,1]^2 : y < y_{\text{crit}}\} \backslash Z; \quad \mathcal{W} = [0,1]^2 \backslash (Z \cup \mathcal{A} \cup \mathcal{B}).$$

▶ **Definition 11** (GK conditions [39]). *Let* $\mu(x,y) = 3f_1(x)f_2(y) + xf_1'(x)f_2(y) + yf_2'(y)f_1(x)$. *Given a canonical partition of* $[0,1]^2$ *induced by* s_1, s_2, *and* P_{crit}, *we say that it satisfies* GK conditions with respect to f_1, f_2 *if it satisfies the following conditions:*

- $\mu(x,y) > 0$ *for all* $(x,y) \in [0,1]^2$, *and*
- $\int_Z \mu(x,y) = 1$, *and*
- $\int_{s_1(y)}^1 \mu(x,y)\mathrm{d}x = f_1(1)f_2(y)$, *for all* $y \in [0, y_{crit}]$, *and*
- $\int_{s_2(x)}^1 \mu(x,y)\mathrm{d}y = f_1(x)f_2(1)$, *for all* $x \in [0, x_{crit}]$.

Finally by [39] (specifically, their Theorem 1 and duality discussions in Section 2.2), we have the following characterization of the optimal mechanism.

▶ **Theorem 12** (Uniqueness and characterization of optimal auction [39]). *Given probability density functions* f_1, f_2 *over* $[0,1]$. *Suppose that canonical partition* $(Z, \mathcal{A}, \mathcal{B}, \mathcal{W})$ *induced by* 1*-Lipschitz, concave and non-increasing functions* s_1, s_2, *and* $P_{\text{crit}} \in [0,2]$ *satisfies the GK conditions. The utility function* $u(x,y)$ *of the optimal IC and IR mechanism for selling two items to a single additive buyer with independent prior distributions* f_1, f_2 *is given by*

$$u(x,y) = \max(0, x - s_1(y), y - s_2(x), x + y - P_{\text{crit}}).$$

Specifically,

- *if* $(x,y) \in Z$, $u(x,y) = 0$;
- *if* $(x,y) \in \mathcal{A}$, $u(x,y) = y - s_2(x)$;
- *if* $(x,y) \in \mathcal{B}$, $u(x,y) = x - s_1(y)$;
- *if* $(x,y) \in \mathcal{W}$, $u(x,y) = x + y - P_{\text{crit}}$.

Moreover, the optimal utility function $u(x,y)$ *is unique in regions* $\mathcal{A}, \mathcal{B}, Z$.

8 Limitations of One-Way Quantum Protocols

[24] studies the optimal auction of selling two items to a single additive buyer with i.i.d. valuations from Beta$(1,2)$. It characterizes the unique utility function $u(\cdot)$ for any optimal mechanism. In particular they show that in the region $y = 1$ and $x \in [0, 0.06]$, the unique utility function for optimal mechanism is given by $u(x,1) = \frac{2-2x}{4-5x}$.

In this section, we show that no finite one-way quantum mechanism can implement this utility function for $x \in [0, 0.06]$ and $y = 1$. It it worth noting that, although only one qubit is exchanged in a single round per definition of our main model, the following negative result applies to any one-way quantum protocol with an arbitrarily large (but finite) message size.

▶ **Theorem 13.** *Given the two-item additive type space $[0,1]^2$, for any non-linear rational function $R(x)$, for any $r > 0$, there is no IC quantum one-way protocol $\mathcal{P}$ has utility function $u(x,1) = R(x)$ for $x \in [0,r)$.*

9 A Quantum One-Way Mechanism for An Uncountable Menu

In this section, we are going to construct an example of a prior over two additive, independent items, where the corresponding optimal auction can: (i) be implemented by a one-way protocol in 1 qubit and 2 classical bits; but (ii) requires an uncountably infinite menu.

▶ **Theorem** (Theorem 3 restated)**.** *For the problem of auctioning two items to a single buyer, there is a Bayesian prior over independent item values, such that there is a revenue-optimal one-way quantum auction protocol where the buyer sends 1 qubit and 2 classical bits; yet no finite classical auction protocol can achieve the optimal revenue.*

At a high level, we construct the example in the following steps:

1. First, we want to apply Theorem 12 to identify the utility function for optimal mechanisms. To simplify the verification of GK conditions, we choose $f_1(x) = 1$, for $x \in [0,1]$, aka the value for item 1 is drawn uniformly from $[0,1]$. By choosing $f_1(x) = 1$, the measure $\mu(x,y)$ defined in GK conditions can be simplified to $\mu(x,y) = 3f_2(y) + yf_2'(y)$.
2. Next, we choose a non-increasing, 1-Lipschitz, concave function s_1. In addition, we require s_1 to be a non-piecewise-linear function, so the utility function in the region $\mathcal{B}$ of the canonical partition $u(x,y) = x - s_1(y)$ is non-piecewise-linear, which implies no finite menu can characterize it. Moreover, as we discussed in the last section, to be able to be implemented by a one-way quantum protocol, $s_1(y)$ has to be a function in the form $\|Ay + B\|$ for some Hermitian matrices A and B. After some trial and error, it turns out that $s_1(y) = \frac{49}{24} - \frac{1}{4}(3y + \sqrt{121/4 - 10y + y^2})$ is a good idea.
3. With the chosen s_1, the next step is to reverse-engineer Theorem 12 to obtain a probability density function $f_2(y)$, function s_2 and critical price P_{crit} such that the canonical partition induced by $s_1, s_2, P_{\text{crit}}$ satisfies GK conditions. In particular, By the third bullet of Definition 11, $f_2(y)$ has to satisfy the following ODE:

$$(1 - s_1(y))\,(3f_2(y) + yf_2'(y)) = f_2(y).$$

4. Finally, we construct a one-way quantum protocol whose utility function is exactly the one given by Theorem 12.

By solving the ODE in bullet 2, we obtain

$$f_2(y) = \frac{c \cdot \left(-6g(y) + 12y + \sqrt{5737} + 5\right)^{-\frac{75}{58} - \frac{7551}{58\sqrt{5737}}} (g(y) - 2y + 10)^3 \left(6g(y) - 12y + \sqrt{5737} - 5\right)^{\frac{7551}{58\sqrt{5737}} - \frac{75}{58}}}{(11 - 2y + g(y))^{99/29}}, \quad (2)$$

where $g(y) = \sqrt{4y^2 - 40y + 121}$, and c is the normalization factor such that

$$\int_0^1 f_2(y)\,\mathrm{d}y = 1.$$

9.1 Optimal Mechanisms

In this subsection, we will further define another non-increasing, 1-Lipschitz, concave function s_2 and critical price $P_{\text{crit}} \in [0,2]$. Next, we verify the canonical partition induced by s_1, s_2, and P_{crit} satisfy GK conditions and give the characterization of the optimal mechanism by applying Theorem 12.

First, one can verify that $\mu(x,y) = 3f_2(y) + yf_2'(y)$ is negative for all $(x,y) \in [0,1]^2$.

Next, given $f_1(x) = 1$, by the last bullet of Definition 11, we set $s_2(x) \approx 0.558$ as a constant function such that $\int_{s_2(x)}^{1}(3f_2(y) + yf_2'(y))\mathrm{d}y = f_2(1)$.

Finally, we set $P_{\text{crit}} \approx 0.669$ such that $\int_Z (3f_2(y)+yf_2'(y)) = 1$. Moreover, line $x+y = P_{\text{crit}}$ intersects both curves $x = s_1(y)$ and $y = s_2(x)$. With s_1, s_2 and P_{crit} we now have $x_{\text{crit}} \approx 0.111$ and $y_{\text{crit}} \approx 0.005$. By definition, the canonical partition induced by s_1, s_2 and P_{crit} satisfies GK conditions. Therefore, by Theorem 12, we give the following characterization of the optimal mechanism for selling two items to a single additive buyer with values independently distributed according to $f_1 = 1$ and f_2 given by (2).

▶ **Lemma 14.** *The optimal mechanism for selling two items to a single additive buyer with values independently distributed according to $f_1 = 1$ and f_2 defined by (2) is given by the utility function*

$$u(x,y) = \max(0, x - s_2(y), y - s_1(x), x + y - P_{\text{crit}}).$$

Specifically, $u(x,y) = x - s_2(y)$ in region B is not a piecewise linear function, and it is the unique utility function for all optimal mechanisms in region B.

9.2 Exact One-Way Quantum Protocol

In this subsection, we give an IC one-way quantum protocol with exactly the same utility as the one characterized in Lemma 14.

Protocol implementation

In the first (and the only) round, the buyer send a single qubit with reduced density matrix ρ, and two classical bits b_1, b_2. If $b_1 = b_2 = 0$, then the seller terminates the protocol with empty allocation and payment 0. If $b_1 = b_2 = 1$, then the seller terminates the protocol with allocation $\{1,2\}$ and payment $P_{\text{crit}} \approx 0.669$. If $b_1 = 0, b_2 = 1$, then the seller terminates the protocol with alloation $\{2\}$ and payment $s_2(0) \approx 0.558$. Finally, if $b_1 = 1, b_2 = 0$, the seller measures the qubit using the following POVM and terminates the protocol with allocation and payment associated with each measurement outcome.

$$A_1 = \begin{pmatrix} 2/5 & \sqrt{21}/16 \\ \sqrt{21}/16 & 1/4 \end{pmatrix}, \quad a_1 = \{1,2\}, \quad p_1 = 2,$$

$$A_2 = \begin{pmatrix} 2/5 & -\sqrt{21}/16 \\ -\sqrt{21}/16 & 1/4 \end{pmatrix}, \quad a_2 = \{1,2\}, \quad p_2 = 0,$$

$$A_3 = \begin{pmatrix} 1/5 & 0 \\ 0 & 0 \end{pmatrix}, \quad a_3 = \{1,2\}, \quad p_3 = \frac{299}{24},$$

$$A_4 = \begin{pmatrix} 0 & 0 \\ 0 & 1/2 \end{pmatrix}, \quad a_4 = \{1\}, \quad p_4 = \frac{7}{12}.$$

We define the buyer's suggested strategy as the optimal response to the seller's strategy, given his private value x and y.

10 (Barely) interactive one-way quantum auction protocols

In Section 8 we see an example of a prior whose optimal mechanism cannot be implemented by a finite one-way quantum auction protocol. In this section, we introduce a barely interactive multi-round quantum auction protocol which is optimal for this example, i.e. $f_1(x) = 2(1-x), f_2(y) = 2(1-y)$ for $(x,y) \in [0,1]^2$.

▶ **Theorem 15** (Theorem 4). *For the problem of auctioning two items to a single buyer, there is a Bayesian prior over independent item values, such that there is a revenue-optimal quantum auction protocol where the seller sends 1 qubit to the buyer, who replies with 1 qubit and 2 classical bits; yet no finite classical or one-way quantum auction protocol can achieve the optimal revenue.*

Due to [24] Section 8.2.1, the (unique) optimal mechanism for this example can be characterized by the following lemma.

Protocol implementation

The seller's strategy is as follows.

In this protocol, the seller first prepares an EPR pair: $\frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle)$ and sends one qubit of the EPR pair to the buyer.

The density matrix of an EPR pair is $\rho_{\text{EPR}} = \begin{pmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix}$.

Next, the seller receives one qubit from the buyer (and two classical bits). We denote by b_1, b_2 the two classical bits. If $b_1 = b_2 = 0$, the protocol terminates with empty allocation and payment 0. If $b_1 = b_2 = 1$ the protocol terminates with allocation $\{1, 2\}$ and payment $P_{\text{crit}} \approx 0.5535$ defined in [24]. If $b_1 \neq b_2$, the seller measures the joint state (two qubits) of his half of the EPR pair and the qubit she receives from the buyer. The seller will use the following POVM and corresponding allocation and payments. For convenience, we define bundle $\pi = \{1\}$ if $b_1 = 1$, and $\pi = \{2\}$ if $b_2 = 1$.

$$A_1 = \begin{pmatrix} \frac{3}{20} & 0 & 0 & \frac{1}{10} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2}{5} & 0 \\ \frac{1}{10} & 0 & 0 & \frac{2}{5} \end{pmatrix}, \quad a_1 = \pi, \quad p_1 = 3,$$

$$A_2 = \begin{pmatrix} \frac{23}{80} & 0 & 0 & -\frac{1}{10} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{3}{5} & 0 \\ -\frac{1}{10} & 0 & 0 & \frac{3}{5} \end{pmatrix}, \quad a_2 = \pi, \quad p_2 = 0,$$

$$A_3 = \begin{pmatrix} \frac{9}{16} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad a_3 = \{1, 2\}, \quad p_3 = 0.$$

We can verify by calculation that this is a valid POVM as all three matrices are positive semidefinite and $A_1 + A_2 + A_3 = I$.

We define the buyer's suggested strategy as the optimal response to the seller's strategy, given his private value x and y.

11 Limitations of finite-round quantum protocols

In this section we give an example where no finite IC and IR protocol obtains optimal revenue.

▶ **Theorem** (Theorem 5 restated). *For the problem of auctioning two items to a single buyer, there is a Bayesian prior over independent item values, such that there is a revenue-optimal classical auction protocol that requires a constant number of bits* in exepctation*; yet no finite quantum auction protocol can achieve the optimal revenue.*

Below are definitions of semialgebraic sets and semialgebraic functions. (See e.g. [23] for reference.)

▶ **Definition 16** (semialgebraic sets). *A subset of $\mathbb{R}^n$ is semialgebraic if it can be represented as a finite union of sets of the form:*

$$\{x \in \mathbb{R}^n : f(x) = 0, g_1(x) > 0, \ldots, g_m(x) > 0\},$$

where f and g_is are real polynomials in x.

▶ **Definition 17** (Semialgebraic functions). *A function $f : \mathbb{R}^n \to \mathbb{R}$ is semialgebraic if its graph $\{(x, y) \in \mathbb{R}^{n+1} : f(x) = y\}$ is a semialgebraic set.*

11.1 The utility function of finite round IC protocol is semialgebraic

▶ **Lemma 18.** *Given an IC an IR finite-round quantum auction protocol, its utility function $u(x)$ is semialgebraic.*

11.2 A mechanism with a non-semialgebraic utility function

[39] characterizes the optimal mechanism for selling two items to an additive buyer with i.i.d. priors $\frac{e^{-x}}{1-1/e}$ (i.e. $f_1(x) = \frac{e^{-x}}{1-1/e}$ and $f_2(y) = \frac{e^{-y}}{1-1/e}$). In particular, they show that the utility function of the optimal mechanism satisfies $u(x, 1) = 2x + W(e^{1-x}(2-x)) - 1$ for $x \in [0, 0.1]$, where $W(\cdot)$ is the Lambert W function[14]. Furthermore, by Theorem 12, we also know this utility function is unique in this region ($y = 1, x \in [0, 0.1]$).

Next, we show that the unique utility function $g(x) = 2x + W(e^{1-x}(2-x)) - 1$ is not a semialgebraic function. Together with Lemma 18, this implies that no finite quantum IC protocol achieves exactly optimal revenue (aka completing the proof of Theorem 5).

▶ **Lemma 19.** *Let $f : \mathbb{R} \to \mathbb{R}$ be a semialgebraic function. $f(x)$ cannot be equal to $g(x) = 2x + W(e^{1-x}(2-x)) - 1$ on $[0, r)$ for any $r > 0$.*

References

1 Andris Ambainis, Leonard J Schulman, Amnon Ta-Shma, Umesh Vazirani, and Avi Wigderson. The quantum communication complexity of sampling. *SIAM Journal on Computing*, 32(6):1570–1585, 2003. doi:10.1137/S009753979935476.

2 Sepehr Assadi. Combinatorial auctions do need modest interaction. In *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, pages 145–162, 2017. doi:10.1145/3033274.3085121.

3 Sepehr Assadi, Hrishikesh Khandeparkar, Raghuvansh R Saxena, and S Matthew Weinberg. Separating the communication complexity of truthful and non-truthful combinatorial auctions. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on theory of computing*, pages 1073–1085, 2020. doi:10.1145/3357713.3384267.

4 Moshe Babaioff, Liad Blumrosen, and Michael Schapira. The communication burden of payment determination. *Games and Economic Behavior*, 77(1):153–167, 2013. doi:10.1016/j.geb.2012.08.007.

[14] Defined as the inverse function of $f(w) = w \cdot e^w$. Moreover, by the Lagrange inversion theorem, W is analytic everywhere on $(-1/e, \infty)$.

5 Moshe Babaioff, Yannai A. Gonczarowski, and Noam Nisan. The menu-size complexity of revenue approximation. *Games Econ. Behav.*, 134:281–307, 2022. doi:10.1016/j.geb.2021.03.001.

6 Moshe Babaioff, Nicole Immorlica, Brendan Lucier, and S. Matthew Weinberg. A simple and approximately optimal mechanism for an additive buyer. *J. ACM*, 67(4):24:1–24:40, 2020. URL: https://dl.acm.org/doi/10.1145/3398745, doi:10.1145/3398745.

7 Yakov Babichenko, Shahar Dobzinski, and Noam Nisan. The communication complexity of local search. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 650–661. ACM, 2019. doi:10.1145/3313276.3316354.

8 Yakov Babichenko and Aviad Rubinstein. Communication complexity of nash equilibrium in potential games (extended abstract). In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1439–1445. IEEE, 2020. doi:10.1109/FOCS46700.2020.00137.

9 Yakov Babichenko and Aviad Rubinstein. Communication complexity of approximate nash equilibria. *Games Econ. Behav.*, 134:376–398, 2022. doi:10.1016/j.geb.2020.07.005.

10 Ziv Bar-Yossef, T. S. Jayram, and Iordanis Kerenidis. Exponential separation of quantum and classical one-way communication complexity. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 128–137, New York, NY, USA, 2004. Association for Computing Machinery. doi:10.1145/1007352.1007379.

11 Salman Beigi, Peter Shor, and John Watrous. Quantum interactive proofs with short messages. *Theory of Computing*, 7, April 2010. doi:10.4086/toc.2011.v007a007.

12 Liad Blumrosen, Noam Nisan, and Ilya Segal. Auctions with severely bounded communication. *J. Artif. Intell. Res.*, 28:233–266, 2007. doi:10.1613/jair.2081.

13 Simina Brânzei and Noam Nisan. Communication complexity of cake cutting. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019.*, page 525, 2019. doi:10.1145/3328526.3329644.

14 Mark Braverman, Jieming Mao, and S. Matthew Weinberg. Interpolating between truthful and non-truthful mechanisms for combinatorial auctions. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1444–1457, 2016. doi:10.1137/1.9781611974331.ch99.

15 Mark Braverman, Jieming Mao, and S. Matthew Weinberg. On simultaneous two-player combinatorial auctions. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2256–2273, 2018. doi:10.1137/1.9781611975031.146.

16 Patrick Briest, Shuchi Chawla, Robert Kleinberg, and S. Matthew Weinberg. Pricing randomized allocations. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 585–597, 2010. doi:10.1137/1.9781611973075.49.

17 Ioannis Caragiannis and Ariel D. Procaccia. Voting almost maximizes social welfare despite limited communication. *Artif. Intell.*, 175(9-10):1655–1671, 2011. doi:10.1016/j.artint.2011.03.005.

18 Xi Chen, Ilias Diakonikolas, Anthi Orfanou, Dimitris Paparas, Xiaorui Sun, and Mihalis Yannakakis. On the complexity of optimal lottery pricing and randomized mechanisms. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1464–1479, 2015. doi:10.1109/FOCS.2015.93.

19 John F Clauser, Michael A Horne, Abner Shimony, and Richard A Holt. Proposed experiment to test local hidden-variable theories. *Physical review letters*, 23(15):880, 1969.

20 Richard Cleve, Peter Hoyer, Benjamin Toner, and John Watrous. Consequences and limits of nonlocal strategies. In *Proceedings of the 19th IEEE Annual Conference on Computational Complexity*, CCC '04, pages 236–249, USA, 2004. IEEE Computer Society. doi:10.1109/CCC.2004.1313847.

21 Vincent Conitzer and Tuomas Sandholm. Communication complexity as a lower bound for learning in games. In *Proceedings of the twenty-first international conference on Machine learning*, page 24. ACM, 2004.

22 Vincent Conitzer and Tuomas Sandholm. Communication complexity of common voting rules. In *Proceedings 6th ACM Conference on Electronic Commerce (EC-2005), Vancouver, BC, Canada, June 5-8, 2005*, pages 78–87, 2005. doi:10.1145/1064009.1064018.

23 Michel Coste. An introduction to semialgebraic geometry, 2000.

24 Constantinos Daskalakis, Alan Deckelbaum, and Christos Tzamos. Strong duality for a multiple-good monopolist. *Econometrica*, 85(3):735–767, 2017.

25 Alan Deckelbaum. Quantum correlated equilibria in classical complete information games. *arXiv preprint*, 2011. arXiv:1101.3380.

26 Shahar Dobzinski. Breaking the logarithmic barrier for truthful combinatorial auctions with submodular bidders. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2016, pages 940–948, New York, NY, USA, 2016. ACM. doi:10.1145/2897518.2897569.

27 Shahar Dobzinski. Computational efficiency requires simple taxation. In *FOCS*, 2016.

28 Shahar Dobzinski, Noam Nisan, and Sigal Oren. Economic efficiency requires interaction. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 233–242, 2014. doi:10.1145/2591796.2591815.

29 Shahar Dobzinski and Shiri Ron. The communication complexity of payment computation. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 933–946. ACM, 2021. doi:10.1145/3406325.3451083.

30 Shahar Dobzinski, Shiri Ron, and Jan Vondrák. On the hardness of dominant strategy mechanism design. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 690–703. ACM, 2022. doi:10.1145/3519935.3520013.

31 Shahar Dobzinski and Jan Vondrák. Communication complexity of combinatorial auctions with submodular valuations. In Sanjeev Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1205–1215. SIAM, 2013. doi:10.1137/1.9781611973105.87.

32 Jens Eisert, Martin Wilkens, and Maciej Lewenstein. Quantum games and quantum strategies. *Phys. Rev. Lett.*, 83:3077–3080, October 1999. doi:10.1103/PhysRevLett.83.3077.

33 Tomer Ezra, Michal Feldman, Eric Neyman, Inbal Talgam-Cohen, and S. Matthew Weinberg. Settling the communication complexity of combinatorial auctions with two subadditive buyers. In *the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2019.

34 Ronald Fadel and Ilya Segal. The communication cost of selfishness. *Journal of Economic Theory*, 144(5):1895–1920, 2009. doi:10.1016/J.JET.2007.09.015.

35 Anat Ganor and Karthik C. S. Communication complexity of correlated equilibrium with small support. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, pages 12:1–12:16, 2018. doi:10.4230/LIPIcs.APPROX-RANDOM.2018.12.

36 Anat Ganor, Karthik C. S., and Dömötör Pálvölgyi. On communication complexity of fixed point computation. *ACM Trans. Economics and Comput.*, 9(4):25:1–25:27, 2021. doi:10.1145/3485004.

37 Dmitry Gavinsky. Classical interaction cannot replace a quantum message. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 95–102, New York, NY, USA, 2008. Association for Computing Machinery. doi:10.1145/1374376.1374393.

38 Dmitry Gavinsky, Julia Kempe, Iordanis Kerenidis, Ran Raz, and Ronald de Wolf. Exponential separation for one-way quantum communication complexity, with applications to cryptography. *SIAM Journal on Computing*, 38(5):1695–1708, 2009. doi:10.1137/070706550.

39 Yiannis Giannakopoulos and Elias Koutsoupias. Selling two goods optimally. *Information and Computation*, 261:432–445, 2018. doi:10.1016/J.IC.2018.02.016.

40 Yannai A. Gonczarowski. Bounding the menu-size of approximately optimal auctions via optimal-transport duality. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 123–131, 2018. doi:10.1145/3188745.3188786.

41 Yannai A. Gonczarowski, Noam Nisan, Rafail Ostrovsky, and Will Rosenbaum. A stable marriage requires communication. *Games Econ. Behav.*, 118:626–647, 2019. doi:10.1016/j.geb.2018.10.013.

42 Yannai A. Gonczarowski and S. Matthew Weinberg. The sample complexity of up-to-ε multidimensional revenue maximization. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, 2018. doi:10.1109/FOCS.2018.00047.

43 Mika Göös and Aviad Rubinstein. Near-optimal communication lower bounds for approximate nash equilibria. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 397–403, 2018. doi:10.1109/FOCS.2018.00045.

44 Chenghao Guo, Zhiyi Huang, and Xinzhi Zhang. Settling the sample complexity of single-parameter revenue maximization. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019.*, pages 662–673, 2019. doi:10.1145/3313276.3316325.

45 Gus Gutoski and John Watrous. Toward a general theory of quantum games. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 565–574. ACM, 2007. doi:10.1145/1250790.1250873.

46 Sergiu Hart and Yishay Mansour. How long to equilibrium? the communication complexity of uncoupled equilibrium procedures. *Games and Economic Behavior*, 69(1):107–126, 2010. doi:10.1016/J.GEB.2007.12.002.

47 Sergiu Hart and Noam Nisan. Selling multiple correlated goods: Revenue maximization and menu-size complexity. *J. Econ. Theory*, 183:991–1029, 2019. doi:10.1016/j.jet.2019.07.006.

48 Sergiu Hart and Philip J. Reny. Maximizing Revenue with Multiple Goods: Nonmonotonicity and Other Observations. *Theoretical Economics*, 10(3):893–922, 2015.

49 Alexander S. Holevo. Bounds for the quantity of information transmitted by a quantum communication channel. *Problemy Peredachi Informatsii*, 9(3), 1973.

50 Rahul Jain, Zhengfeng Ji, Sarvagya Upadhyay, and John Watrous. Qip = pspace. *Communications of the ACM*, 53(12):102–109, 2010. doi:10.1145/1859204.1859231.

51 Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. Mip* = re. *Commun. ACM*, 64(11):131–138, October 2021. doi:10.1145/3485628.

52 Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. In *the 3rd Annual ACM Conference on Electronic Commerce (EC)*, 2001.

53 Hagay Levin, Michael Schapira, and Aviv Zohar. Interdomain routing and games. *SIAM J. Comput.*, 40(6):1892–1912, 2011. doi:10.1137/080734017.

54 A. M. Manelli and D. R. Vincent. Multidimensional Mechanism Design: Revenue Maximization and the Multiple-Good Monopoly. *Journal of Economic Theory*, 137(1):153–185, 2007. doi:10.1016/J.JET.2006.12.007.

55 A. M. Manelli and D. R. Vincent. Bayesian and Dominant-Strrategy Implementation in the Independent Private-Values Model. *Econometrica*, 78(6):1905–1938, 2010.

56 David A Meyer. Quantum games and quantum algorithms. *arXiv preprint*, 2000. arXiv:quant-ph/0004092.

57 Ashley Montanaro. Quantum states cannot be transmitted efficiently classically. *Quantum*, 3:154, 2019.

58 Ashley Montanaro and Changpeng Shao. Quantum communication complexity of linear regression. *arXiv preprint arXiv:2210.01601*, 2022. doi:10.48550/arXiv.2210.01601.

59 Anand Natarajan and John Wright. Neexp is contained in mip. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 510–518. IEEE, 2019. doi:10.1109/FOCS.2019.00039.

60 Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*, volume 2. Cambridge university press Cambridge, 2001.

61 Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting prices. *J. Economic Theory*, 129(1):192–224, 2006. doi:10.1016/j.jet.2004.10.007.

62 Gregory Pavlov. Optimal mechanism for selling two goods. *The B.E. Journal of Theoretical Economics*, 11(3), 2011.

63 Benjamin Plaut and Tim Roughgarden. Communication complexity of discrete fair division. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2014–2033, 2019. doi:10.1137/1.9781611975482.122.

64 Ariel D. Procaccia and Jeffrey S. Rosenschein. The communication complexity of coalition formation among autonomous agents. In *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006*, pages 505–512, 2006. doi:10.1145/1160633.1160727.

65 Ran Raz. Exponential separation of quantum and classical communication complexity. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, STOC '99, pages 358–367, New York, NY, USA, 1999. Association for Computing Machinery. doi:10.1145/301250.301343.

66 Oded Regev and Bo'az Klartag. Quantum one-way communication can be exponentially stronger than classical communication. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 31–40, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/1993636.1993642.

67 Jean-Charles Rochet and Philippe Chone. Ironing, sweeping, and multidimensional screening. *Econometrica*, 66(4):783–826, 1998.

68 Tim Roughgarden and Omri Weinstein. On the communication complexity of approximate fixed points. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 23, page 55, 2016.

69 Aviad Rubinstein, Raghuvansh R. Saxena, Clayton Thomas, S. Matthew Weinberg, and Junyao Zhao. Exponential communication separations between notions of selfishness. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 947–960. ACM, 2021. doi:10.1145/3406325.3451127.

70 Aviad Rubinstein and Junyao Zhao. The randomized communication complexity of optimal randomized auctions. In *Symposium on Theory of Computing, STOC 2021*. ACM, 2021.

71 Travis C. Service and Julie A. Adams. Communication complexity of approximating voting rules. In *International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, Valencia, Spain, June 4-8, 2012 (3 Volumes)*, pages 593–602, 2012. URL: http://dl.acm.org/citation.cfm?id=2343781.

72 Hao Tang, Boning Li, Guoqing Wang, Haowei Xu, Changhao Li, Ariel Barr, Paola Cappellaro, and Ju Li. Communication-efficient quantum algorithm for distributed machine learning. *arXiv preprint*, 2022. arXiv:2209.04888.

73 John Thanassoulis. Haggling over substitutes. *Journal of Economic Theory*, 117:217–245, 2004. doi:10.1016/J.JET.2003.09.002.

74 John Watrous. Pspace has constant-round quantum interactive proof systems. *Theoretical Computer Science*, 292(3):575–588, 2003. doi:10.1016/S0304-3975(01)00375-9.

75 Zhaohui Wei and Shengyu Zhang. Full characterization of quantum correlated equilibria. *Quantum Inf. Comput.*, 13(9-10):846–860, 2013. doi:10.26421/QIC13.9-10-7.

76 S. Matthew Weinberg and Zixin Zhou. Optimal multi-dimensional mechanisms are not locally-implementable. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, EC '22, pages 875–896, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3490486.3538334.

77 A Chi-Chih Yao. Quantum circuit complexity. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, pages 352–361. IEEE, 1993. doi:10.1109/SFCS.1993.366852.

Quantum 2-SAT on Low Dimensional Systems Is QMA_1-Complete: Direct Embeddings and Black-Box Simulation

Dorian Rudolph[1]
Department of Computer Science and Institute for Photonic Quantum Systems (PhoQS), Paderborn University, Germany

Sevag Gharibian
Department of Computer Science and Institute for Photonic Quantum Systems (PhoQS), Paderborn University, Germany

Daniel Nagaj
Institute of Physics, Slovak Academy of Sciences, Bratislava, Slovakia

Abstract

Despite the fundamental role the Quantum Satisfiability (QSAT) problem has played in quantum complexity theory, a central question remains open: At which local dimension does the complexity of QSAT transition from "easy" to "hard"? Here, we study QSAT with each constraint acting on a d_A-dimensional and d_B-dimensional qudit pair, denoted $(d_A \times d_B)$-QSAT. Our first main result shows that, surprisingly, QSAT on qubits can remain QMA_1-hard, in that (2×5)-QSAT is QMA_1-complete. (QMA_1 is a quantum analogue of MA with perfect completeness.) In contrast, (2×2)-QSAT (i.e. Quantum 2-SAT on qu*b*its) is well-known to be poly-time solvable [Bravyi, 2006]. Our second main result proves that $(3\times d)$-QSAT on the 1D line with $d \in O(1)$ is also QMA_1-hard. Finally, we initiate the study of $(2\times d)$-QSAT on the 1D line by giving a frustration-free 1D Hamiltonian with a unique, entangled ground state.

As implied by our title, our first result uses a *direct embedding*: We combine a novel clock construction with the 2D circuit-to-Hamiltonian construction of [Gosset and Nagaj, 2013]. Of note is a new simplified and *analytic* proof for the latter (as opposed to a partially numeric proof in [GN13]). This exploits Unitary Labelled Graphs [Bausch, Cubitt, Ozols, 2017] together with a new "Nullspace Connection Lemma", allowing us to break low energy analyses into small patches of projectors, and to improve the soundness analysis of [GN13] from $\Omega(1/T^6)$ to $\Omega(1/T^2)$, for T the number of gates. Our second result goes via *black-box* reduction: Given an *arbitrary* 1D Hamiltonian H on d'-dimensional qudits, we show how to embed it into an effective 1D $(3\times d)$-QSAT instance, for $d \in O(1)$. Our approach may be viewed as a weaker notion of "analog simulation" (*à la* [Bravyi, Hastings 2017], [Cubitt, Montanaro, Piddock 2018]). As far as we are aware, this gives the first "black-box simulation"-based QMA_1-hardness result.

2012 ACM Subject Classification Theory of computation → Complexity classes; Theory of computation → Quantum complexity theory

Keywords and phrases quantum complexity theory, Quantum Merlin Arthur (QMA), Quantum Satisfiability Problem (QSAT), Hamiltonian simulation

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.85

Related Version *Full Version*: `https://arxiv.org/abs/2206.05243` [40]

Funding *Dorian Rudolph*: Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) project 432788384

[1] corresponding author



© Dorian Rudolph, Sevag Gharibian, and Daniel Nagaj;
licensed under Creative Commons License CC-BY 4.0



Sevag Gharibian: Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) projects 450041824 and 432788384. BMBF project PhoQuant (grant number 13N16103). Project "PhoQC" from the programme "Profilbildung 2020", an initiative of the Ministry of Culture and Science of the State of North Rhine-Westphalia.
Daniel Nagaj: Projects APVV-22-0570 (DeQHOST), and VEGA 2/0183/21 (DESCOM).

Acknowledgements We thank Jonas Kamminga for proofreading, and Libor Caha for helpful discussions.

1 Introduction

Boolean satisfiability problems have long served as a testbed for probing the boundary between "easy" (i.e. poly-time solvable) and "hard" (e.g. NP-complete) computational problems. A striking early example of this is the fact that while 3-SAT is NP-complete [12, 33, 28], 2-SAT is in P [38, 15, 31, 18, 5, 36]. Despite this, *MAX*-2-SAT (i.e. what is the *maximum* number of satisfiable clauses of a 2-CNF formula?) remains NP-complete [20]! Thus, the border between tractable and intractable can often be intricate, hiding abrupt transitions in complexity.

In the quantum setting, generalizations of k-SAT and MAX-k-SAT have similarly played a central role, additionally due to their strong physical motivation. The input here is a *k-local Hamiltonian* $H = \sum_i H_i$ acting on n qubits, which is a $2^n \times 2^n$ complex Hermitian matrix (a quantum analogue of a Boolean formula on n bits), given via a succinct description $\{H_i\}$. Here, each H_i is a $2^k \times 2^k$ operator acting on[2] k out of n qubits (i.e. a local quantum clause). Given H, the goal is to compute the smallest eigenvalue $\lambda_{\min}(H)$ of H, known as the *ground state energy*. The corresponding eigenvector, in turn, is the *ground state*. This k-local Hamiltonian problem (k-LH) generalizes MAX-k-SAT, and formalizes the question: If a many-body quantum system is cooled to near absolute zero, what energy level will the system relax into? The complexity of k-LH is well-understood, and analogous to MAX-2-SAT, even 2-LH is complete for[3] Quantum Merlin-Arthur (QMA) [30, 29, 14].

The quantum generalization of k-SAT (as opposed to *MAX*-k-SAT), on the other hand, is generally *less* understood. In contrast to k-LH, one now asks whether there exists a ground state of H which is simultaneously a ground state for *each local term* H_i (analogous to a string satisfying *all* clauses of a k-SAT formula). Formally, in Quantum k-SAT (k-QSAT), each H_i is now a projector, and one asks whether $\lambda_{\min}(H) = 0$. As in the classical setting, it is known that the locality k leads to a complexity transition: On the one hand, Gosset and Nagaj [24] proved that 3-QSAT is[4] QMA$_1$-complete, while on the other hand, Bravyi gave [8] a poly-time classical algorithm for 2-QSAT (in fact, 2-QSAT is solvable in linear time [4, 16]).

Systems of higher local dimension. In the quantum setting, however, there is an additional, physically motivated direction to probe for complexity transitions for 2-QSAT – *systems of higher local dimension*. Perhaps the most striking example of this is that, while Boolean satisfiability problems in 1D are efficiently solvable via dynamic programming (even for

[2] Formally, if H_i acts on a subset $S \subseteq [n]$ of qubits, to make the dimensions match we consider $(H_i)_S \otimes I_{[n]\setminus S}$.
[3] QMA is the bounded-error quantum analogue of NP, now with poly-size quantum proof and quantum verifier [30].
[4] QMA$_1$ is QMA but with perfect completeness; see Definition 2.1. Note that while MA = MA$_1$ [43], whether QMA$_1$ = QMA remains a major open question.

Table 1 Summary of the known complexity results for $(a\times b)$-QSAT. Rows denote values of d_A, columns values of d_B. New results from this work are bold.

	2	3	4	5	6	7
2	$\in$ P [8]	NP-hard [34]	NP-hard [34]	**$\mathbf{QMA_1}$-comp**	**$\mathbf{QMA_1}$-comp**	$\cdots$
3		NP-hard [34]	**$\mathbf{QMA_1}$-comp**	QMA_1-comp [17]	QMA_1-comp [17]	$\cdots$
4			**$\mathbf{QMA_1}$-comp**	QMA_1-comp [17]	QMA_1-comp [17]	$\cdots$
5				QMA_1-comp [17]	QMA_1-comp [17]	$\cdots$
6					$\cdots$	$\cdots$

d-level systems instead of bits), Aharonov, Gottesman, Irani and Kempe showed [2] that 2-LH on the line remains QMA-complete, *so long as* one uses local dimension $d = 12$! An analogous result for 2-QSAT with $d = 11$ was subsequently given by Nagaj [34]. This raises the guiding question of this work:

What is the smallest local dimension that can encode a QMA_1*-hard problem?*

There are three results we are aware of here. Define $(d_A\times d_B)$-QSAT as 2-QSAT where each constraint acts on a qu-d_A-it and a qu-d_B-it, i.e. on $\mathbb{C}^{d_A}\otimes\mathbb{C}^{d_B}$. (When $d_A \neq d_B$, for this to be well-defined, the interaction graph of the instance must be bipartite.) Not only is 2-QSAT on qubits (i.e. (2×2)-QSAT) in P, Chen, Chen, Duan, Ji, and Zheng showed [11] that a YES instance always has a witness that is a product of single- and two-qubit states. On the other hand, Bravyi, Caha, Movassagh, Nagaj, and Shor gave a frustration-free[5] qutrit construction (i.e. on local dimension $d = 3$) on the 1D line[6] [9] with a *unique*, *entangled* ground state. While this construction does not encode a computation (and thus does not give QMA_1-hardness), it is an important first step in that it shows even such low dimensional systems can encode entangled witnesses (entanglement is necessary, otherwise an NP witness is possible). Together, these works [11] and [9] suggest that qu*b*it systems are a no-go barrier for QMA_1-hardness. Prior to these, Eldar and Regev [17] came closest to establishing a result about qubit systems, showing that (3×5)-QSAT is QMA_1-hard (on general graphs).[7] But the key question remains open – *Can qubit systems support* QMA_1*-hardness for Quantum* 2*-SAT, i.e. is* $(2\times d)$-QSAT QMA_1*-hard for some* $d \in O(1)$*?*

Our results. We show two main results, along with a third preliminary one.

1. QMA_1*-hardness for qubit systems.* The complexity of $(d_A\times d_B)$-QSAT including our new results is summarized in Table 1. The first main result is as follows.

▶ **Theorem 1.1.** (2×5)-QSAT *is* QMA_1*-complete with soundness* $\Omega(1/N^2)$.

Here, N denotes the number of gates[8] used by the QMA_1-verifier. Thus, qubit systems *can* encode QMA_1-hardness. Let us be clear that the surprising part of this is *not* that this setting is intractable – indeed, even classical (2×3)-SAT is known to be NP-hard (e.g. [34]).

[5] A *frustration-free* Hamiltonian is a YES instance of QSAT, i.e. a local Hamiltonian $H \succeq 0$ with $\lambda_{\min}(H) = 0$.

[6] "On the line" means $H = \sum_{i=1}^{m} H_{i,i+1}$, i.e. the qudits can be depicted as a sequence with each consecutive constraint acting on the next pair of qudits in the sequence.

[7] QMA_1-hardness of (4×9)-QSAT is claimed without proof in a footnote in [35] (see Reference 12 therein).

[8] As an aside, we assume in this paper that the QMA_1 verifier utilizes the standard "Clifford + T" gate set. See also [39] for an in-depth discussion of the "gate set issue" of QMA_1.

What *is* surprising is that one can encode *quantum* verifications in such low dimension, for two reasons. First, *a priori* a 2-dimensional space for 2-local constraints appears too limited to *exactly*[9] encode a computation – a two-dimensional space only appears to suffice to encode a "data qubit", so where does one encode the "clock" tracking the computation? Second, the entanglement of $(2\times d)$ systems is generally easier to characterize than that of $(d\times d)$ systems. For example, whether a (2×2) or (2×3) system is entangled is detectable via Peres' Positive Partial Transpose (PPT) criterion [37], whereas detecting entanglement for $(d\times d)$ systems (for polynomial d) becomes strongly NP-hard [25, 27, 21]. And recall that a "sufficiently entangled" ground space is necessary to encode QMA$_1$-hardness.

As a complementary result, we show that one can "trade" dimensions in the construction above if one is careful, i.e. the 5 in (2×5) can be reduced to 4 at the expense of increasing 2 to 3.

▶ **Theorem 1.2.** (3×4)-QSAT *is* QMA$_1$*-complete with soundness* $\Omega(1/N^2)$.

One can also consider general Quantum 2-SAT on qu-d-its of identical dimension d, which is QMA$_1$-complete for $d = 5$ [17]. Theorem 1.2 improves this to $d = 4$.

We remark that obtaining Theorems 1.1 and 1.2 is significantly more involved than the previous best QMA$_1$-hardness for (3×5)-QSAT [17] (details in "Techniques" below), and in particular requires (among other ideas) a simplified analysis of the advanced 2D-Hamiltonian framework of Gosset and Nagaj for $(2\times 2\times 2)$-QSAT (i.e. 3-QSAT on qubits) [24][10], a new clock construction, and the Unitary Labelled Graphs of Bausch, Cubitt and Ozols [6]. One of the payoffs is that we obtain a "tight"[11] soundness gap of $\Omega(1/N^2)$ for Theorem 1.1, compared to the $\Omega(1/N^6)$ gap of [24]. This allows us to recover the 3-QSAT hardness results of [24], but with improved soundness[12]:

▶ **Theorem 1.3.** 3-QSAT *is* QMA$_1$*-complete with soundness* $\Omega(1/N^2)$.

2. Low dimensional systems on the line. Our next main result is the following.

▶ **Theorem 1.4.** $(3\times d)$-QSAT *on a line is* QMA$_1$*-complete with* $d = O(1)$.

This improves significantly on the previous best 1D (11×11)-QSAT QMA$_1$-hardness construction of Nagaj [34], showing that frustration free Hamiltonians on qutrits can encode not just entangled ground states (*cf.* [9]), but also QMA$_1$-hard computations.

For clarity, there is a trade-off in the parameter, d, which we now elaborate. A key novelty of Theorem 1.4 is its proof via *black-box* reduction (see "Techniques" below), as opposed to a direct embedding of a QMA$_1$ computation. Specifically, given an *arbitrary* 1D Hamiltonian H on d'-dimensional qudits, we embed it into an effective 1D $(3\times d)$-QSAT instance with $d \in O((d')^4)$ (i.e. $d \in O(1)$ for constant d'). On the not-so-positive side, the generality of

[9] An *exact* encoding appears needed by definition of QSAT. This is in strong contrast to 2-LH, where *approximate* encodings are allowed (since all constraints need not be simultaneously satisfiable) via perturbation theory [29, 10].

[10] At a first glance, it seems like $(2\times 2\times 2)$-QSAT can be reinterpreted as (2×4)-QSAT-QSAT by combining two qubits into a qu-4-dit. Indeed, a *single* $(2\times 2\times 2)$-constraint can be embedded straightforwardly into a (2×4)-constraint. However, the interactions between a *set* of $(2\times 2\times 2)$-constraints seems difficult to directly reproduce via a set of (2×4)-constraints.

[11] By "tight", we mean that it is not currently known for either QSAT or LH how to get a promise gap larger than $\Omega(1/N^2)$ [42].

[12] For clarity, this improved soundness is not necessary for our results, but an added bonus of our new analysis.

this approach means that an input 1D Hamiltonian H on d'-dimensional qudits is mapped to a 1D Hamiltonian on constraints of dimension $(3 \times O((d')^4))$, i.e. the first dimension drops to 3 at the expense of the second dimension increasing. Thus, for example, if we plug in the 1D (11×11) QMA_1-hardness construction [34], Theorem 1.4 gives QMA_1-hardness for (2×76176)-QSAT.

On the positive side, however, our technique is the first use of (a weaker notion) of the influential idea of *local simulation* (Bravyi and Hastings [7], Cubitt, Montanaro, and Piddock [13]; see Definition 4.1 of [22] for a simpler statement) in the study of QSAT. Roughly, in such simulations, given a local Hamiltonian H on n qubits, one typically applies a local isometry V to each qubit, i.e. maps $H \mapsto V^{\otimes n} H (V^\dagger)^{\otimes n}$, blowing up the input space A into a larger, "logical" space B. By cleverly choosing an appropriate Hamiltonian H' on B, one forces the low-energy space of H' to approximate H. Traditionally, the drawback of this approach is its reliance on perturbation theory, which necessarily gives rise to[13] *frustrated* Hamiltonians H'. Here, however, we show for the first time that a weaker[14] version of such local embeddings can be designed even for the frustration-free setting, ultimately yielding Theorem 1.4.

3. Towards qubits on the line. Finally, we initiate the study (2×d) on a line by proving that even a frustration-free Hamiltonian on a line of alternating particles with dimensions 2 and 4 can have a unique entangled ground state.

▶ **Theorem 1.5.** *Consider a line of $2n$ particles such that the i-th particle has dimension 2 for even i and 4 for odd i. There exists a Hamiltonian $H = \sum_{i=1}^{n} A_{2i-1,2i} + \sum_{i=1}^{n-1} B_{2i,2i+1} + L_{1,2} + R_{2n-1,2n}$, where A, B, L, R are 2-local projectors, such that the nullspace of H is $\mathcal{N}(H) = \mathrm{Span}\{|\psi\rangle\}$ and $|\psi\rangle$ is entangled across all cuts.*[15]

Techniques. We focus on our main results, Theorem 1.1 and Theorem 1.4.

QMA_1*-hardness for* (2×5)-QSAT. The basic idea behind most (if not all) QMA_1-hardness constructions for QSAT is to embed a so-called *history state* $|\psi_{\mathrm{hist}}\rangle$ encoding the QMA_1 verifier's circuit U into the nullspace of a local Hamiltonian H. Specifically, let $U = U_T \cdots U_1$ be a sequence of 1- and 2-qubit gates. Then, the Feynman-Kitaev history state [19, 30] is given by

$$|\psi_{\mathrm{hist}}\rangle = \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} \big(U_t \cdots U_1 |\psi_{\mathrm{proof}}\rangle_A |0 \cdots 0\rangle_B \big) |t\rangle_C, \tag{1}$$

for register A the proof register, B the ancilla register, and C the clock register. This should be thought of as the quantum analogue of a Cook-Levin tableau, where the steps of the computation are now encoded in *superposition* over time t. While this basic idea is simple, the challenge lies in implementing it when restricted to certain local dimensions, clause/interaction localities, or interaction geometries. Progress for QSAT in particular has largely been difficult, since in contrast to the setting of LH, one does not have access to the tools of perturbation theory, which necessarily create Hamiltonians with frustrated ground spaces (i.e. empty nullspaces). Instead, QSAT hardness constructions must carefully stitch together rather involved gadgets to logically effect the history state idea above.

[13] In words, perturbation theory is only known to be able to show QMA-hardness for k-LH, as opposed to QMA_1-hardness for k-QSAT.

[14] For clarity, the simulations of [7, 13] reproduce the whole target Hamiltonian H, whereas our approach is weaker in that we prove simulation of only H's nullspace. A similar notion of simulation is discussed by Aharonov and Zhou [3], called *gap simulation*, where only the ground space is approximately simulated.

[15] The Schmidt rank is $\Theta(1)$, but we do not explicitly analyze it here.

To this end, we begin by discussing our approach for Theorem 1.1. We wish to encode local Hamiltonian constraints which force any potential null state to repeatedly update the current time in the clock register from t to $t+1$, along with applying the tth gate of U, as sketched above. However, even embedding a clock (never mind the full computation of U!) into the nullspace of a $(2\times d)$-Hamiltonian is non-trivial, as we only have qubit-systems at our disposal to act as "auxiliary particles" (as opposed to qutrits in [17]). Thus, our first challenge is to break this "qutrit barrier". We illustrate our techniques in a toy example.

Our starting point is to embed a simple clock into the null-space of a (3×3)-Hamiltonian, and subsequently logically simulate this clock by pairing "indicator qubits" (i.e. qubit auxiliary particles) with 6-dimensional qudits as follows. It is straightforward to write down a (3×3)-Hamiltonian encoding the simple sequence of clock states (e.g. on 3 qutrits) $(|100\rangle, |200\rangle, |210\rangle, |220\rangle, |221\rangle, |222\rangle)$ in its nullspace, where note that only a single qutrit changes in each step. To break the "qutrit barrier", we thus now implement each qutrit in this clock *logically* in a larger space consisting of a 6-dimensional qudit and a triple of "indicator qubits". Formally, qutrit $|x\rangle \in \{|0\rangle, |1\rangle, |2\rangle\}$ is encoded via

$$|x\rangle \mapsto |x\rangle_\alpha |000\rangle_\beta + |x'\rangle_\alpha |0^x 10^{2-x}\rangle_\beta \quad \text{for } x \in \{0,1,2\} \text{ and } x' = x+3 \in \{3,4,5\}, \tag{2}$$

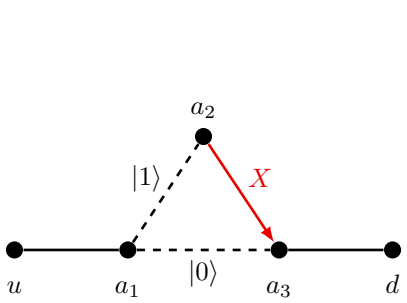
where α is 6-dimensional, and β consists of three qubits labelled $\beta_0, \beta_1, \beta_2$. To illustrate the idea behind this encoding, consider a projector that enforces equality (i.e. applies an identity gate) between the first two clock states (we also call this a $|100\rangle \leftrightarrow |200\rangle$ transition). In (3×3), we can use the projector onto $|10\rangle - |20\rangle$ (i.e. $\frac{1}{2}(|1\rangle - |2\rangle)(\langle 1| - \langle 2|) \otimes |0\rangle\langle 0|$) acting on the first two qutrits, where $|0\rangle\langle 0|$ on the second qutrit ensures that the projector only acts on the first two timesteps. Using our encoding, we can realize this transition with a (2×6)-projector onto $|11\rangle_{\alpha,\beta_0} - |21\rangle_{\alpha,\beta_0}$. This uses two features of the encoding: (1) a transition on a logical qutrit can be implemented with a 1-local projector on qudit α (here $\frac{1}{2}(|1\rangle - |2\rangle)(\langle 1| - \langle 2|)_\alpha$), and (2) a projection onto a standard basis state $|i\rangle \in \{|0\rangle, |1\rangle, |2\rangle\}$ of a logical qudit can be implemented with a 1-local projector onto an indicator qubit β_i (here $|1\rangle\langle 1|_{\beta_i}$).

This qutrit encoding (Equation (2)) generalizes to qudits of any dimension, and combined with the (3×5)-Hamiltonian of [17], gives QMA$_1$-hardness of (2×10)-QSAT. We can remove some unused indicator qubits to improve this to (2×7)-QSAT (see [40]) as the indicator requires an extra qudit dimension ($|x\rangle$ and $|x'\rangle$).[16]

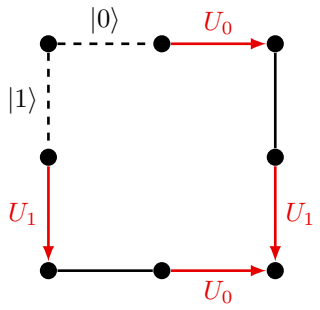
Further improvement to (2×5)-QSAT requires much more work. In order to directly enforce the application of a gate $U_t \neq \mathbb{I}$ in a history state (Equation (1)) with a 2-local projector, we can only use a single qudit from the proof register and a single qudit from the clock register. Hence, we need to be able to project onto clock states with a 1-local qudit projector. For that, Eldar and Regev [17] introduce the notion of "*u*nborn", "*d*ead", and "*a*live" states, where non-trivial transitions only happen on the "alive" states, which uniquely identify a single timestep. Having separate "unborn" and "dead" states allows the clock states to have the general structure $d \cdots d\, a\, u \cdots u$, which is easy to enforce with 2-local constraints.[17] In fact, [17] uses three "alive states" a_1, a_2, a_3 to implement a CNOT gate via a "triangle gadget", which routes states with $|0\rangle$ and $|1\rangle$ in the control register along two different paths, only applying the X gate on the path of $|1\rangle$ (see Figure 1.1a).

[16] We need both and $|x\rangle_\alpha$ and $|x'\rangle_\alpha$ so that logical $|x\rangle$ and $|y\rangle$ differ only in a single qudit (on a subspace), which allows for a 1-local transition. Since the indicator qubits are "off" on $|x\rangle_\alpha$, we need a unique $|x'\rangle_\alpha$ to "turn on" the correct indicator via a transition.

[17] A natural question is if we can reduce the alphabet $\{a, u, d\}$ to a simpler binary alphabet $\{0, 1\}$, so that



(a) Effective implementation of a CNOT gate from u to d. Conditioned on a control bit being $|1\rangle$ (respectively $|0\rangle$), the gadget takes transition $|a_1\rangle \leftrightarrow |a_2\rangle$ (respectively $|a_1\rangle \leftrightarrow |a_3\rangle$). Only the former path applies the X gate (indicated by edge $|a_2\rangle \to |a_3\rangle$).



(b) In this case, when the control bit is $|0\rangle$ (respectively $|1\rangle$), U_1U_0 (respectively U_0U_1) is applied. A suitable choice of non-commuting U_0, U_1 implements CNOT (up to 1-local unitaries) from top left to bottom right. In contrast to the triangle gadget in (a), this gadget in (b) is only the "effective Hamiltonian" and the actual construction is more involved due to geometric limitations.

Figure 1.1 Graphical representation of the 2-local gate gadgets of (a) [17], and (b) [24] as well as this work. Vertices represent clock states, black (solid) edges identity transitions, red (directed) edges unitary transitions, dashed edges conditional identity transitions.

Gosset and Nagaj [24] (see Figure 1.1b) introduce the idea of a 2D-clock with two separate clock registers for the X- and Y-directions. Hence, clock states are vertices in a 2D-grid and therefore can realize separate paths while only requiring transitions between "adjacent" (in one dimension) clock states. The next idea of our construction is to implement this 2D-Hamiltonian of [24] using just two "alive" states in 2-QSAT (compared with three "alive" states in [17] for 2-QSAT; note [24] uses their 2D-clock for 3-QSAT). Figure 1.1b depicts the "effective Hamiltonian" leveraging this idea in order to implement a CNOT gate. Note Figure 1.1b is simplified; the actual implementation with 2-local projectors is significantly more involved (see full version [40]). This is due to complications which arise from geometric limitations: For example, we are not able to restrict the U_0 transition (Figure 1.1b) in the Y-direction (i.e. U_0 would be applied from $x = 2$ to $x = 3$ for all coordinates y).

To implement the ideas of the previous paragraph, we extract the 2D-Hamiltonian of [24] as a generic construction into which we can plug any clock satisfying certain requirements (see Theorem 3.1). Thereby we obtain QMA_1-hardness for (3×4)-QSAT. Then we simulate this (3×4)-clock on a (2×5)-system via the indicator qubit principle (similar to Equation (2)). Note that when implementing a logical 4-dit on a (2×5)-system, we can only use a single indicator qubit. Thus, we need a very carefully crafted (3×4)-clock, and further "technical tricks", which we omit in this introduction.

Finally, to analyze the 2D-Hamiltonian, we prove a novel technical lemma, which we dub the "Nullspace Connection Lemma" (Lemma 3.2). This enables us to split the 2D-Hamiltonian into smaller gadgets (see Figure 1.1b and [40] for the individual gadgets), each of which implements a small part of the history state and is relatively straightforward to analyze. The gadgets are then connected with additional transitions to form the complete Hamiltonian (see [40]), whose nullspace is spanned by superpositions of the gadget's history

clock states look like $\{|100\rangle, |010\rangle, |001\rangle\}$. Unfortunately, there is no set of 2-local projectors whose joint nullspace is the span of $\{|100\rangle, |010\rangle, |001\rangle\}$. The common unary encoding $|100\rangle, |110\rangle, |111\rangle$ cannot be used here because such states cannot be identified 1-locally.

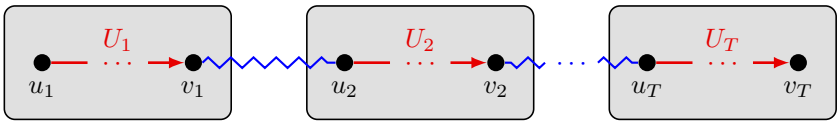


Figure 1.2 Nullspace Connection Lemma: Each box represents a gadget that only acts on a subset of clock states (vertices). Each gadget has an input vertex u_i, and an output vertex v_i. Its nullspace is spanned by history states applying gate U_i from u_i to v_i. Blue zigzag edges connect outputs v_i to inputs u_{i+1} with identity transitions. We show that the entire Hamiltonian has spectral gap $\Omega(1/N^2)$[18] and its nullspace is spanned by history states applying $U_T \cdots U_2 U_1$ from u_1 to v_T.

states. Figure 1.2 gives a visualization and informal statement of the lemma. The Connection Lemma is quite generic and can also be applied to, e.g., the original circuit Hamiltonian of Kitaev [30] (see Remark 3.3), matching the $\Omega(1/N^2)$ soundness (smallest non-zero eigenvalue) therein. Furthermore, the Connection Lemma is proven directly via the Geometric Lemma [30] and does not require transformation to the Laplacian matrix of a random walk, unlike [30]. Our main application of the Connection Lemma is to give a simplified proof for the 2D-Hamiltonian with improved soundness. Since the Connection Lemma requires modifications to the Hamiltonian of [24], we actually use our own variant of [24]'s gadget, which is slightly more compact (using $6M+1$ clock states as opposed to $9M+3$ for M gates). Finally, we do not rely on numerical methods to derive the nullspaces of the individual gadgets (*cf.* the gadget analysis of [24], which required numerics), and instead give a formal proof based on the theory of *unitary labeled graphs* [6]. The upshot is that our overall approach is very flexible – by combining unitary labelled graphs with the Connection Lemma, one can in principle analyze combinations of Hamiltonian gadgets beyond just our 2D setting with relative ease. Therefore, we believe the Connection Lemma will find uses in future works.

QMA$_1$*-hardness for* $(3\times d)$-QSAT *on the line.* As mentioned earlier, in contrast to the direct embedding for Theorem 1.1, for Theorem 1.4 we instead use a black-box simulation. In other words, we do not give a new circuit-to-Hamiltonian mapping, but instead bootstrap the prior QMA$_1$-completeness result of QSAT on a line of qu-d-its with $d = 11$ due to Nagaj [34]. Specifically, we construct a general embedding of *any* 1D-Hamiltonian H on a line of qudits into a Hamiltonian H' on an alternating line of qu-d'-its and qutrits. We then plug the [34] construction into this embedding. To design our embedding, each qu-d'-it is treated as two logical qu-d''-its with $d'' = \sqrt{d'}$ (see Figure 1.3; Figure 4.2 for technical details). We construct a Hamiltonian H'_{logical} that restricts the $(d''\times 3\times d'')$ systems to a d-dimensional subspace, which acts as a logical qu-d-it. This logical subspace has the key feature that, in a sense, the logical qu-d-it can be "accessed" from either its left or right qu-d''-it. This allows us to encode the 2-local terms of H as 1-local terms acting on the qu-d'-its of H'. The rough idea behind the logical subspace is to treat the $(d''\times 3\times d'')$ system as three bins capable of holding two types of balls (red and black) that are exchanged between the bins to move the information between the left and right qu-d''-it in each triple $(d''\times 3\times d'')$. When one d''-bin is full, the other d''-bin must be empty and the total number i of red balls encodes the basis state $|i/2\rangle$ of the corresponding logical qudit (Figure 4.1 depicts the balls and bins). To encode a d-dimensional system, this requires $O(d)$ red balls and $O(d)$ black balls, so that $d'' = \Theta(d^2)$, and thus $d' = \Theta(d^4)$.

[18] This gap is for gadgets of constant size, which suffice for our purposes. However, Lemma 3.2 is more general and even allows gadgets of non-constant size.

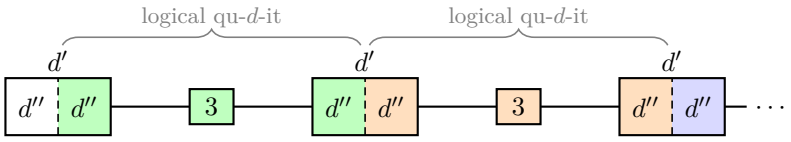


Figure 1.3 Simulation of $(d\times d)$-QSAT with $(3\times d')$-QSAT on the line. Qu-d'-its are split into two qu-d''-its with $d'' = \sqrt{d'}$. Each $(d''\times 3\times d'')$ system implements a logical qu-d-it.

As previously described, our embedding can be seen as a weaker notion of *simulation* in the sense of [7, 13], in that formally our embedding is achieved via application of *local isometries* (i.e. local observables are mapped to local observables), followed by additional constraints on the logical space (Equations (19a) and (19b)). However, in contrast to [7, 13], we do not analyze the structure of higher energy spaces of H', and only show that H' preserves the nullspace of H as well as its smallest non-zero eigenvalue, up to a linear factor (see Lemma 4.3), as this suffices for our purposes.

Open questions. Although we have shown that qubit systems can support QMA_1-hard problems, the frontier for characterizing the complexity transition of local Hamiltonian problems from low to high local dimension remains challenging. In our setting, in particular, the main open question is whether one can obtain QMA_1-hardness even for (2×3)-QSAT? This would complete the complexity characterization for $(d_A\times d_B)$-QSAT, as recall (2×2)-QSAT (i.e. 2-QSAT) is in P [8]. Getting this down to (2×3)-QSAT (even (3×3)-QSAT or (2×4)-QSAT), however, appears difficult, requiring ideas beyond those introduced here.

As for the 1D line, the best hardness results for 2-LH and 2-QSAT are on 8-dimensional [26] and 11-dimensional [34] systems, respectively. Is 1D 2-QSAT on qudits with $2 < d < 11$ QMA_1-hard? We showed that 1D $(3\times d)$-QSAT is QMA_1-hard, but due to our black-box approach we only get $d = 76176$. So it seems likely that this d can be improved significantly. Perhaps more interesting is the question whether 1D $(2\times d)$-QSAT is still QMA_1-hard. We showed that even 1D (2×4)-dimensional constraints can support a unique globally entangled ground state (Theorem 1.5), but this construction alone does not embed a computation, and thus does not yield QMA_1-hardness. Can this be bootstrapped to obtain QMA_1-hardness for 1D $(2\times d)$-QSAT? For 1D 2-LH, the situation is even worse – on qubits, these systems can only be efficiently solved in the presence of a constant spectral gap [32]. In contrast, for inverse polynomial gap, and even with the promise of an NP witness (i.e. via Matrix Product State), the problem is NP-complete [41]. What is the complexity of 1D 2-LH on qudits with $2 < d < 8$?

Organization. Section 2 gives formal definitions for QMA_1, $(d_A\times d_B)$-QSAT, and states the Geometric Lemma with various corollaries. Section 3 proves our main result, the QMA_1-completeness of (2×5)-QSAT. Section 4 proves the QMA_1-completeness of $(3\times d)$-QSAT on a line. Section 5 gives the construction of the (2×4)-Hamiltonian on a line with entangled ground space.

2 Preliminaries

In this section, we formally introduce QMA_1 and elaborate on the Geometric Lemma.

2.1 QMA$_1$

The complexity class QMA$_1$ is defined in the same way as QMA, but with the additional requirement of *perfect completeness*, i.e., in the YES-case, there exists a proof that the verifier accepts with a probability of exactly 1. Consequently, QMA$_1$ is not known to be independent of the gate set [24], as approximate decompositions of arbitrary unitaries generally breaks perfect completeness. Therefore, we have to fix a gate set before we define QMA$_1$, and here we follow [24] in choosing the "Clifford + T" gate set $\mathcal{G} = \{\widehat{H}, T, \mathrm{CNOT}\}$, where $\widehat{H}$ denotes the Hadamard gate. Giles and Selinger [23] have proven that a unitary can be synthesized exactly with gate set $\mathcal{G}$ iff its entries are in the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$.

▶ **Definition 2.1** (QMA$_1$). *A promise problem $A = (A_{\mathrm{yes}}, A_{\mathrm{no}})$ is in* QMA$_1$ *if there exists a poly-time uniform family of quantum circuits $\{Q_x\}$ with gate set $\mathcal{G}$ such that:*

- *(Completeness) If $x \in A_{\mathrm{yes}}$, then there exists a proof $|\psi\rangle$ with $\Pr[Q_x \text{ accepts } |\psi\rangle] = 1$.*
- *(Soundness) If $x \in A_{\mathrm{no}}$, then then for all proofs $|\psi\rangle$, $\Pr[Q_x \text{ accepts } |\psi\rangle] \leq 1 - 1/\operatorname{poly}(|x|)$.*

▶ **Definition 2.2** ($(d_A \times d_B)$-QSAT). *Consider a system of d_A- and d_B-dimensional particles, denoted $A_i, i \in [n_A]$ and $B_j, j \in [n_B]$, respectively. In the* $(d_A \times d_B)$-QSAT *problem, the input is a $(d_A \times d_B)$-Hamiltonian $H = \sum_{i \in [n_A], j \in [n_B]} \Pi_{A_i, B_j}$ with 2-local projectors Π_{A_i, B_j} acting non-trivially only on particles A_i and B_j. Decide:*

- *(YES) $\lambda_{\min}(H) = 0$.*
- *(NO) $\lambda_{\min}(H) \geq 1/\operatorname{poly}(n_A + n_B)$.*

Note that the projectors of $(d_A \times d_B)$-QSAT need to have a specific form such that the problem is contained in QMA$_1$ with our chosen gate set (see Section 3.4).

2.2 Geometric Lemma

In our proofs, we frequently apply Kitaev's Geometric Lemma [30] as well as its extension to the frustration-free case due to Gosset and Nagaj [24], where we are interested in lower-bounding the smallest non-zero eigenvalue of a Hamiltonian H, denoted $\gamma(H)$. In the following, we also give further refinements of these statements. As in [24], we use the notation $H|_S = \Pi_S H \Pi_S$ for the restriction of the Hamiltonian H to the subspace S, where Π_S is the projector onto S.[19] $\mathcal{N}(H)$ denotes the nullspace of H.

▶ **Lemma 2.3** (Kitaev's Geometric Lemma [30] as stated in [24]). *Let $H = H_A + H_B$ with $H_A \succeq 0$ and $H_B \succeq 0$. Let $S = \mathcal{N}(H_A)$ and Π_B be the projector onto $\mathcal{N}(H_B)$. Suppose $\mathcal{N}(H) = \{0\}$. Then $\gamma(H) \geq \min\{\gamma(H_A), \gamma(H_B)\} \cdot (1 - \sqrt{c})$, where $c = \max_{|v\rangle \in S : \langle v|v\rangle = 1} \langle v|\Pi_B|v\rangle$.*

▶ **Corollary 2.4** ([24, Corollary 1]). *Let $H = H_A + H_B$ where $H_A \succeq 0$ and $H_B \succeq 0$ each have nonempty nullspaces. Let Γ be the subspace of states in $\mathcal{N}(H_A)$ that are orthogonal to $\mathcal{N}(H)$, and let Π_B be the projector onto $\mathcal{N}(H_B)$. Then $\gamma(H) \geq \min\{\gamma(H_A), \gamma(H_B)\} \cdot (1 - \sqrt{d})$, where $d = \|\Pi_B|_\Gamma\| = \max_{|v\rangle \in \Gamma : \langle v|v\rangle = 1} \langle v|\Pi_B|v\rangle$.*

We give a slightly tighter statement of [24, Corollary 2][20]:

▶ **Corollary 2.5.** *Let $H = H_A + H_B$ where $H_A \succeq 0$ and $H_B \succeq 0$ each have nonempty nullspaces. Let $S = \mathcal{N}(H_A)$ and suppose $H_B|_S$ is not the zero matrix. Then*

$$\gamma(H) \geq \min\{\gamma(H_A), \gamma(H_B)\} \cdot \frac{\gamma(H_B|_S)}{2\|H_B|_S\|}. \tag{3}$$

[19] Note, this is not the standard restriction of linear map to a subspace since H does not necessarily map S to itself.

[20] The only difference is that we have $\|H_B|_S\|$ instead of $\|H_B\|$ in the denominator.

Proof. As stated in [24], $\gamma(H_B|_S) = \min_{v\in\Gamma:\langle v|v\rangle=1}\langle v|H_B|v\rangle$ with Γ as defined in Corollary 2.4. Hence for all unit $|v\rangle \in \Gamma \subseteq S$,

$$\gamma(H_B|_S) \leq \langle v|H_B|v\rangle \leq \langle v|(\mathbb{I}-\Pi_B)|v\rangle\|H_B|_S\| \tag{4}$$

and thus $d \leq 1-\gamma(H_B|_S)/\|H_B|_S\|$. The statement then follows from $1-\sqrt{1-x} \geq \frac{x}{2}$ for $x \in [0,1]$. ◀

▶ **Corollary 2.6.** *Let $H = H_A + H_B$ where $H_A \succeq 0$ and $H_B \succeq 0$ each have nonempty nullspaces. Let $S = \mathcal{N}(H_A)$ and suppose $H_B|_S$ is not the zero matrix. Then*

$$\gamma(H) \geq \min\{\gamma(H_A),\gamma(H_B)\} \cdot \min_{|v\rangle\in\Gamma:\langle v|v\rangle=1} \langle v|(\mathbb{I}-\Pi_B)|v\rangle/2. \tag{5}$$

Proof. Follows from Corollary 2.4 and the fact $1-\sqrt{1-x} \geq \frac{x}{2}$ for $x \in [0,1]$. ◀

▶ **Corollary 2.7.** *Let $H_A \succeq 0, H_B \succeq 0$ be Hamiltonians with $\mathcal{N}(H_A) \subseteq \mathcal{N}(H_B)$. Then $\gamma(H_A+H_B) \geq \min\{\gamma(H_A),\gamma(H_B)\}$.*

Proof. In Corollary 2.4, we have $\Gamma = \mathcal{N}(H_A)\cap\mathcal{N}(H_A+H_B)^\perp = \mathcal{N}(H_A)\cap\mathcal{N}(H_A)^\perp = \{0\}$. Thus $d = \|\Pi_B|_\Gamma\| = 0$ and $\gamma(H_A+H_B) \geq \min\{\gamma(H_A),\gamma(H_B)\}$. ◀

3 (2×5)-QSAT is QMA$_1$-complete

Our proof mainly leverages the 2D-Hamiltonian construction of Gosset and Nagaj [24], which gives a circuit-to-Hamiltonian embedding with a set of "simple" projectors that are constructed from a suitable clock encoding. In Section 3.2, we give the main technical tool which we use to analyze the soundness of the aforementioned Hamiltonian, and in Section 3.3 we give our clock construction for (2×5)-QSAT.

3.1 2D-Hamiltonian

The following theorem extracts the 2D-Hamiltonian construction central to [24] so that we can use it in conjunction with our own clock construction. In the full version [40], we give a complete proof with a simplified Hamiltonian construction and improved analysis that gives soundness $\Omega(\gamma(H^{(N)}_{\text{clock}})/N^2)$, as opposed to $\Omega(\gamma(H^{(N)}_{\text{clock}})/N^6)$ from [24]. Our proof is fully analytic, improving on the partially numeric analysis of [24].

Note, our construction is structurally almost the same as [24], which would also work in conjunction with our clock (see Section 3.3), requiring only a slight modification to the H_{init} and H_{end} terms. However, the application of Lemma 3.2 (necessary for improved soundness) to their Hamiltonian is not as straightforward because there is no separate gadget for 1-local gates. So, our contribution to the following theorem are a simplified proof and improved soundness bounds. The intuition of the construction is given in the introduction.

▶ **Theorem 3.1.** *Suppose we are given Hamiltonian terms as follows:*

(1) *Clock Hamiltonian $H^{(N)}_{\text{clock}}$ acting on Hilbert space $\mathcal{H}_{\text{clock}}$ with nullspace $\mathcal{N}(H^{(N)}_{\text{clock}}) = \text{Span}\{|C_1\rangle,\ldots,|C_n\rangle\} =: \mathcal{C}$ for* clock states *$|C_i\rangle$.*

(2) *Projectors $h_{i,i+1}(U)$ acting on $\mathcal{H}_{\text{comp}}\otimes\mathcal{H}_{\text{clock}}$, where U is a unitary acting on $\mathcal{H}_{\text{comp}}$, such that*

$$\begin{aligned}\left(\mathbb{I}\otimes\Pi^{(N)}_{\text{clock}}\right) h_{i,i+1}(U) \left(\mathbb{I}\otimes\Pi^{(N)}_{\text{clock}}\right) = c_1\big((\mathbb{I}\otimes|C_i\rangle\langle C_i| + \mathbb{I}\otimes|C_{i+1}\rangle\langle C_{i+1}|)\\ -(U^\dagger\otimes|C_i\rangle\langle C_{i+1}| + U\otimes|C_{i+1}\rangle\langle C_i|)\big)\end{aligned} \tag{6}$$

for constant[21] *c_1. We write $h_{i,i+1} := h_{i,i+1}(\mathbb{I})$.*

[21] It suffices for our purposes to have the same c for all i. In principle, however, one can also allow different

(3) *Projectors $C_{\geq i}, C_{\leq i}$ acting on $\mathcal{H}_{\text{clock}}$ such that*

$$\Pi^{(N)}_{\text{clock}} C_{\geq i} \Pi^{(N)}_{\text{clock}} = \sum_{j=i}^{N} c_{2,i,j} |C_j\rangle\langle C_j|, \qquad \Pi^{(N)}_{\text{clock}} C_{\leq i} \Pi^{(N)}_{\text{clock}} = \sum_{j=1}^{i} c_{3,i,j} |C_j\rangle\langle C_j|, \tag{7}$$

and $c_{2,i,j}, c_{3,i,j} \geq c_2$ for all $i, j \in [N]$ for some constant c_2.

(4) *All above projectors $(h_{i,i+1}(U), C_{\geq i}, C_{\leq i})$ pairwise commute, besides $h_{i,i+1}(U)$ and $h_{i,i+1}(U')$ for non-commuting U, U'. For all $i \neq j$, $h_{i,i+1}(U) h_{j,j+1}(U') = 0$.*

(5) *If $\Pi_1, \ldots, \Pi_k$ are projectors of the form $C_{\leq i}, C_{\geq i}$, then $\langle C_{j_1}|\Pi_1 \cdots \Pi_k|C_{j_2}\rangle = 0$ for $j_1 \neq j_2$.*

Then any problem in QMA$_1$ *can be reduced to* QSAT *restricted to Hamiltonians H acting on $\mathcal{H}_{\text{comp}} \otimes \mathcal{H}_{\text{clock},1} \otimes \mathcal{H}_{\text{clock},2}$ (operators acting on these spaces are labeled with subscripts Z, X, Y respectively) with terms $(H^{(N)}_{\text{clock}})_X$, $(H^{(N)}_{\text{clock}})_Y$, $\Pi_Z \otimes (h_{i,i+1})_A$, $(C_{\sim j})_A \otimes (h_{i,i+1})_B$, $(h_{i,i+1}(U))_{Z,A}$, $(C_{\leq i})_A \otimes (C_{\geq j})_B$, where $A, B \in \{X, Y\}, A \neq B$, $\sim \in \{\leq, \geq\}$[22], $i, j \in [N]$, $\Pi_Z \in \{|0\rangle\langle 0|, |1\rangle\langle 1|\}$ is a single-qubit projector acting on $\mathcal{H}_{\text{comp}}$, and U is either a 1-local gate from the* QMA$_1$ *circuit or $U \in \{\sigma^Z, B\}$ with $B = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & i \\ i & 1 \end{smallmatrix}\right)$, $\sigma^Z = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$. The soundness is $\Omega(\gamma(H^{(N)}_{\text{clock}})/N^2)$, where $N = \Theta(g)$ and g is the number of gates used by the* QMA$_1$ *verifier.*

Proof. A rough intuition is given in Section 1 and the formal proof in the full version [40]. ◀

Note, this theorem is not explicitly stated in [24], but is implicitly proven, albeit with a soundness of $\Omega(\gamma(H^{(N)}_{\text{clock}}/N^6))$. Thus, as an immediate first consequence we can (using the clock Hamiltonian of [24]) recover QMA$_1$-hardness of 3-QSAT, but with improved soundness:

▶ **Theorem 1.3.** 3-QSAT *is* QMA$_1$*-complete with soundness $\Omega(1/N^2)$.*

Unfortunately its generality makes Theorem 3.1 somewhat difficult to parse. Hence, we sketch its application to the clock of Kitaev's circuit Hamiltonian [30] as a toy example:

(1) Clock states are given by $|C_i\rangle = |1^i 0^{N-i}\rangle$ with $H_{\text{clock}} = \sum_{i=1}^{N-1} |01\rangle\langle 01|_{i,i+1} + |0\rangle\langle 0|_1$.

(2) $h_{i,i+1}(U)$ essentially applies the gate U from timestep $|C_i\rangle$ to $|C_{i+1}\rangle$. Define for $i \in [N-1]$, $h_{i,i+1}(U) = (\mathbb{I} \otimes |100\rangle\langle 100|_{i,i+1,i+2} + \mathbb{I} \otimes |110\rangle\langle 110|) - (U^\dagger \otimes |100\rangle\langle 110| + U \otimes |110\rangle\langle 100|)$. Equation (6) can easily be verified with $c_1 = 1$. In our actual construction we get $c_1 < 1$ because the clock states are superpositions of multiple standard basis states, of which $h_{i,i+1}$ only acts on one (see Section 3.3).

(3) $C_{\leq i}, C_{\geq i}$ just project onto timesteps $\leq i$ or $\geq i$. Define $C_{\leq i} = |0\rangle\langle 0|_{i-1}$ and $C_{\geq i} = |1\rangle\langle 1|_i$. Equation (7) can easily be verified with $c_2 = 1$.

(4) and **(5)** are only used to prove the soundness lower bound and may be dropped, decreasing soundness by a polynomial factor.

Thus, Theorem 3.1 gives QMA$_1$-completeness of 4-QSAT when applied to Kitaev's unary clock.

3.2 Nullspace Connection Lemma

Abstractly, the 2D-Hamiltonian consists of a sequence of gadgets connected together via the blue zigzag edges, as depicted in Figure 1.2. If we remove the zigzag edges, we can analyze the nullspace and gap of the Hamiltonian easily, as gadgets have only constant size and act on orthogonal clock spaces. One can show that the nullspace of each gadget is spanned by history states on the local clock spaces. The "Nullspace Connection Lemma" below then argues that these local history states can be connected via the zigzag edges.

---

constants depending on the choice of index i.

[22] Here we use "$\sim$" as a placeholder for a relation "$\leq$" or "$\geq$".

Our statement is quite general, so that Lemma 3.2 can be applied more broadly. Due to the decomposition into the local gadgets acting on disjoint sets of clock states, one only needs to compute the nullspaces of constant-size gadgets before applying Lemma 3.2. See Remark 3.3 for an application to Kitaev's circuit Hamiltonian. For instance, Rudolph uses Lemma 3.2 to prove correctness of various circuit-to-Hamiltonian embeddings [39] and computes the gadget nullspaces algebraically. In the full version of this work, we compute nullspaces of the 2D-Hamiltonian gadgets via the theory of Unitary Labelled Graphs [6].

▶ **Lemma 3.2** ("Nullspace Connection Lemma"). *Let*

(1) $K = K_1 \cup \ldots \cup K_m = [N]$ *be a disjoint partition of the indices of* N *clock states with* $u_i, v_i \in K_i, u_i \neq v_i$ *for all* $i \in [m]$.

(2) $H_1 = \sum_{i=1}^{m} H_{1,i}$ *be a Hamiltonian on ancilla space* A *and clock space* C*, such that for all* $i \in [m]$*:*

- **(a)** $\mathcal{N}(H_{1,i}|_{\mathcal{K}_i}) = \mathrm{Span}\{|\psi_i(\alpha_j)\rangle \mid j \in [d]\}$*, where* $\mathcal{K}_i = \mathbb{C}^d_A \otimes \mathrm{Span}\{|v\rangle_C \mid v \in K_i\}$*, and* $|\alpha_1\rangle, \ldots, |\alpha_d\rangle$ *is an orthonormal basis of the ancilla space,*
- **(b)** *the local history states* $|\psi_i(\alpha)\rangle$ *are linear in the input* $|\alpha\rangle$*, i.e., there exists a linear map* L_i *with* $L_i|\alpha\rangle = |\psi_i(\alpha)\rangle$ *and* $L_i^\dagger L_i = \lambda_i \mathbb{I}$ *for some constant* λ_i*,*
- **(c)** H_i *has support only on clock space* $\mathcal{K}_i$*,*
- **(d)** $\||\psi_i(\alpha)\rangle\|^2 =: \delta_i \in [1, \Delta]$*,*
- **(e)** $(\mathbb{I}_A \otimes \langle u_i|_C)|\psi_i(\alpha)\rangle = |\alpha\rangle_A$*,*
- **(f)** $(\mathbb{I}_A \otimes \langle v_i|_C)|\psi_i(\alpha)\rangle = U_i|\alpha\rangle_A$ *for some unitary* U_i*.*

(3) $H_2 = \sum_{i=1}^{m-1} h_{v_i,u_{i+1}}(V_i)$ *with* $h_{v_i,u_{i+1}}(V_i) = \mathbb{I}\otimes|v_i\rangle\langle v_i| + \mathbb{I}\otimes|u_{i+1}\rangle\langle u_{i+1}| - V_i^\dagger \otimes |v_i\rangle\langle u_{i+1}| - V_i \otimes |u_{i+1}\rangle\langle v_i|$ *for unitaries* V_i*.*

(4) $|\alpha_{ij}\rangle = V_{i-1}U_{i-1}\cdots V_1U_1|\alpha_j\rangle$*.*

Then for $H = H_1 + H_2$*,* $\mathcal{N}(H) = \mathrm{Span}\{\sum_{i=1}^{m}|\psi_i(\alpha_{ij})\rangle \mid j \in [d]\}$*,* $\gamma(H) = \Omega(\gamma(H_1)/(m^2\Delta))$*.*

Prior to the proof, let us briefly give an intuitive explanation of the parts. (1) just gives a decomposition of the indices of clock states into disjoint components K_i. One can think of these as a generalization of timesteps to a graph, like in the Unitary Labelled Graphs [6]. (2) is the Hamiltonian corresponding to the unitary gadgets without the connection (zigzag) edges. (a) says that the i-th gadget should apply gate U_i, such that it has a "local history state" from input (e) to output (f). The local history state is linear in the "input state" (b). (c) requires the gadgets to only act on their clock states K_i and thus commute. (d) constrains the norm of history states, which is used to compute the gap. (3) is the Hamiltonian corresponding to the connection edges. (4) describes the input states to each gadget.

Proof. Since the $H_{1,i}$ Hamiltonians act on different clock spaces, they commute and it holds $S := \mathcal{N}(H_1) = \mathrm{Span}\{|\psi_i(\alpha_j)\rangle \mid i \in [m], j \in [d]\}$. Next, we derive $\mathcal{N}(H) = S \cap \mathcal{N}(H_2)$. Partition S into orthogonal subspaces $S_1, \ldots, S_d$, where $S_j = \mathrm{Span}\{|\psi_i(\alpha_{ij})\rangle \mid i \in [m]\}$. Orthogonality holds because $\langle\psi_i(\alpha)|\psi_i(\alpha')\rangle = \langle\alpha|L_i^\dagger L_i|\alpha'\rangle = \lambda_i\langle\alpha|\alpha'\rangle$, and $|\psi_i(\alpha)\rangle \in \mathcal{K}_i$, where $\mathcal{K}_i \subseteq \mathcal{K}_{i'}^\perp$ for $i \neq i'$. As H_2 is block diagonal across $S_1, \ldots, S_d$, it suffices to consider the $\mathcal{N}(H_2|_{S_j})$ individually.

Let $|\psi\rangle = \sum_{i=1}^{m} a_i|\psi_i(\alpha_{ij})\rangle \in S_j$. Then $\langle\psi|H|\psi\rangle = \langle\psi|H_2|\psi\rangle = \sum_{i=1}^{m-1}\langle\psi|h_{v_i,u_{i+1}}(V_i)|\psi\rangle$ with

$$\begin{aligned} h_{v_i,u_{i+1}}(V_i)|\psi\rangle &= a_iU_i|\alpha_{ij}\rangle|v_i\rangle + a_{i+1}|\alpha_{i+1,j}\rangle|u_{i+1}\rangle - a_{i+1}V_i^\dagger|\alpha_{i+1,j}\rangle|v_i\rangle - a_iV_iU_i|\alpha_{ij}\rangle|u_{i+1}\rangle \\ &= a_iU_i|\alpha_{ij}\rangle|v_i\rangle + a_{i+1}|\alpha_{i+1,j}\rangle|u_{i+1}\rangle - a_{i+1}U_i|\alpha_{ij}\rangle|v_i\rangle - a_i|\alpha_{i+1,j}\rangle|u_{i+1}\rangle. \end{aligned}$$

Thus, $\langle\psi|h_{v_i,u_{i+1}}(V_i)|\psi\rangle = a_i^*a_i + a_{i+1}^*a_{i+1} - a_i^*a_{i+1} - a_{i+1}^*a_i = |a_i - a_{i+1}|^2$ and $\mathcal{N}(H_2|_{S_j}) = \mathrm{Span}\{\sum_{i=1}^{m}|\psi_i(\alpha_{ij})\rangle\}$.



Figure 3.1 Applying the Nullspace Connection Lemma to Kitaev's circuit Hamiltonian.

Next, we apply Corollary 2.5 to show $\gamma(H) \geq \min\{\gamma(H_1), \gamma(H_2)\} \cdot \gamma(H_2|_S)/(2\|H_2\|)$. The terms $\|H_2\|, \gamma(H_2)$ are constant as the $h_{v_i,u_{i+1}}$ projectors act on distinct clock states. Hence, $\gamma(H) = \Omega(\gamma(H_1)\gamma(H_2|_S))$. Since $H_2|_S$ is block diagonal, we have $\gamma(H_2|_S) \geq \min_{j\in[d]} \gamma(H_2|_{S_j})$, where $\gamma(H_2|_{S_j}) = \min_{|v\rangle\in\Gamma_j:\langle v|v\rangle=1} \langle v|H_2|v\rangle$ and $\Gamma_j = S_j \cap \mathcal{N}(H)^\perp$.

Let $|\psi\rangle = \sum_{i=1}^m a_i|\psi_i(\alpha_{ij})\rangle \in \Gamma_j, \langle\psi|\psi\rangle = 1$ and $|\phi\rangle = \sum_{i=1}^m |\psi_i(\alpha_{ij})\rangle \in \mathcal{N}(H)$. Then $0 = \langle\phi|\psi\rangle = \sum_{i=1}^m a_i\delta_i$ and $\langle\psi|H_2|\psi\rangle = \sum_{i=1}^{m-1} |a_i - a_{i+1}|^2 \geq (\sum_{i=1}^{m-1} |a_i - a_{i+1}|)^2/m$. Also, there exists an l such that $|a_l|^2\delta_l \geq 1/m$. Via a global rotation, we may assume without loss of generality $a_l > 0$. Since $\sum_{i=1}^m a_i\delta_i = 0$, there must exist k with $\Re(a_k) < 0$. Thus $|a_l - a_k| = |a_l\delta_l - a_k\delta_l|/\delta_l > |a_l\delta_l|/\delta_l \geq 1/\sqrt{m\Delta}$. By the triangle inequality, $\langle\psi|H_2|\psi\rangle \geq 1/(m^2\Delta)$. ◀

▶ Remark 3.3. The Nullspace Connection Lemma can also used to prove correctness of Kitaev's original circuit-to-Hamiltonian construction [30] directly without transforming H_{prop} to a random walk. Recall $H_{\text{prop}} = \sum_{j=2}^N H^j_{\text{prop}}$ with

$$H^j_{\text{prop}} = -\frac{1}{2}(U_j)_A \otimes |j\rangle\langle j-1|_C - \frac{1}{2}(U_j^\dagger)_A \otimes +\frac{1}{2}I_A \otimes (|j\rangle\langle j| + |j-1\rangle\langle j-1|)_C,$$

where A denotes the ancilla space where the computation takes place, and C the clock space. Assume that $U_j = \mathbb{I}$ for even j. Figure 3.1 then depicts H_{prop} for $N = 7$. The graph is only a line since H_{prop} uses an ordinary (single) clock. To apply Lemma 3.2, let $K_1 = \{0, 1\}, K_2 = \{2, 3\}, K_3 = \{4, 5\}, K_4 = \{6, 7\}$, $H_1 = \sum_{j\in\{1,3,5,7\}} H^j_{\text{prop}}$ (red edges in Figure 3.1), and $H_2 = \sum_{j\in\{2,4,6\}} H^j_{\text{prop}}$ (blue zigzag edges in Figure 3.1). It is now easy to verify that all the conditions of Lemma 3.2 are satisfied: For example, $\mathcal{N}(H^1_{\text{prop}}|_{\mathcal{K}_1}) = \text{Span}\{|\psi_1(\alpha_j)\rangle \mid j \in [d]\}$, for $|\psi_1(\alpha_j)\rangle = |\alpha_j\rangle_A|0\rangle_C + U_1|\alpha_j\rangle_A|1\rangle_C$. It follows that $\mathcal{N}(H_{\text{prop}}) = \text{Span}\{|\psi_{\text{hist}}(\alpha_j)\rangle \mid j \in [d]\}$, where $|\psi_{\text{hist}}(\alpha)\rangle = \sum_{i=0}^N U_i \cdots U_1|\alpha\rangle|i\rangle_C$.

The Hamiltonian H_{in} is then used to restrict the ancilla space to $|0\rangle$ on timestep 0, so that $\mathcal{N}(H_{\text{in}} + H_{\text{prop}}) = \text{Span}\{|\psi_{\text{hist}}(0)\rangle\}$.

3.3 Clock Hamiltonian

In this section, we define the clock to prove the QMA$_1$-completeness of (2×5)-QSAT (Theorem 1.1) and (4×3)-QSAT (Theorem 1.2) using Theorem 3.1. The main difficulty was the construction of a suitable clock for Theorem 3.1 using just (2×5)-projectors. The hardness of (3×4)-QSAT is obtained almost for free as part of the proof. Since our construction may seem somewhat arbitrary, we give an intuition in the appendix of [40] by sketching a proof for the QMA$_1$-hardness of (2×7)-QSAT.

We begin by defining several logical states. A 5-dit (whose standard basis states are labeled u, a_1, a_2, d, u') and a qubit are combined to construct a logical 4-dit (labeled $\mathbf{U}, \mathbf{A}_1, \mathbf{A}_2, \mathbf{D}$), as shown on the left of Equation (8) (normalization factors omitted). A 5-dit (labeled u, a, d, x, d') and two qubits are combined to construct a logical qutrit (labeled $\mathbf{u}, \mathbf{a}, \mathbf{d}$), as shown on the right of Equation (8). The labels u, a, d maybe be interpreted as "unborn", "alive", and "dead", respectively, following the convention of [1, 8, 17].

$$\begin{aligned} |\mathbf{U}\rangle &= |u,0\rangle + |u',1\rangle & |\mathbf{u}\rangle &= |u,1,0\rangle + |x,0,0\rangle \\ |\mathbf{A}_1\rangle &= |a_1,0\rangle & |\mathbf{a}\rangle &= |a,0,0\rangle + |x,1,0\rangle \\ |\mathbf{A}_2\rangle &= |a_2,0\rangle & |\mathbf{d}\rangle &= |d,0,0\rangle + |d',0,1\rangle \\ |\mathbf{D}\rangle &= |d,0\rangle \end{aligned} \tag{8}$$

▶ **Lemma 3.4.** *There exist 2-local Hamiltonians $H_{\text{logical}_4}, H_{\text{logical}_3}$ of $(2,5)$-projectors with $\mathcal{N}(H_{\text{logical}_4}) = \text{Span}\{|\mathbf{U}\rangle, |\mathbf{A}_1\rangle, |\mathbf{A}_2\rangle, |\mathbf{D}\rangle\}$ and $\mathcal{N}(H_{\text{logical}_4}) = \text{Span}\{|\mathbf{U}\rangle, |\mathbf{A}_1\rangle, |\mathbf{A}_2\rangle, |\mathbf{D}\rangle\}$.*

Proof. In the full version [40]. ◀

The main "feature" of these logical qudits is the fact that a 1-local qubit projector suffices to identify the states $|\mathbf{U}\rangle$, $|\mathbf{d}\rangle$, "$|\mathbf{u}\rangle$ or $|\mathbf{a}\rangle$" (e.g. $|u\rangle\langle u|$), and implement a transition between $|\mathbf{u}\rangle, |\mathbf{a}\rangle$ as $(|0\rangle - |1\rangle)(\langle 0| - \langle 1|)$ (assuming $|\mathbf{d}\rangle$ is already penalized by another projector).

We make use of these properties in the definition of the *clock states* $|C_i\rangle =: |C_{i,1}\rangle + |C_{i,2}\rangle + |C_{i,3}\rangle + |C_{i,4}\rangle$ for $i \in [N]$, where the states $|C_{i,j}\rangle$ are defined as follows:

$$|C_{i,1}\rangle = |\mathbf{dD}\rangle^{\otimes i-2} \otimes |\mathbf{dA}_2, \mathbf{uU}\rangle \otimes |\mathbf{uU}\rangle^{\otimes N-i} \tag{9a}$$
$$|C_{i,2}\rangle = |\mathbf{dD}\rangle^{\otimes i-2} \otimes |\mathbf{dD}, \mathbf{aU}\rangle \otimes |\mathbf{uU}\rangle^{\otimes N-i} \tag{9b}$$
$$|C_{i,3}\rangle = |\mathbf{dD}\rangle^{\otimes i-2} \otimes |\mathbf{dD}, \mathbf{dU}\rangle \otimes |\mathbf{uU}\rangle^{\otimes N-i} \tag{9c}$$
$$|C_{i,4}\rangle = |\mathbf{dD}\rangle^{\otimes i-2} \otimes |\mathbf{dD}, \mathbf{dA}_1\rangle \otimes |\mathbf{uU}\rangle^{\otimes N-i} \tag{9d}$$

Here, the qudits of the clock register are subdivided into groups of five, where the first three (labeled α, β, γ) represent a logical qutrit and the the last two (labeled δ, ε) represent a logical 4-qudit (e.g. the i-th group of (9d) is $|\mathbf{dA}_1\rangle$). With these clock states, a 1-local transition from $|\mathbf{A}_1\rangle$ of $|C_{i,4}\rangle$ to $|\mathbf{A}_2\rangle$ of $|C_{i+1,1}\rangle$ can be implemented on a single 5-dit: $(|a_1\rangle - |a_2\rangle)(\langle a_1| - \langle a_2|)_{\delta_i}|_{\mathcal{C}} = \Omega(1) \cdot (|C_i\rangle - |C_{i+1}\rangle)(\langle C_i| - \langle C_{i+1}|)$ with $\mathcal{C}$ as below.

▶ **Lemma 3.5.** *$\mathcal{N}(H_{\text{clock}}) = \text{Span}\{|C_1\rangle, \ldots, |C_N\rangle\} =: \mathcal{C}$ and $\gamma(H_{\text{clock}}) = 1$.*

Proof sketch. The first step is to enforce the "logical space" using the Hamiltonian $H_{\text{logical}} = \sum_{i=0}^{N-1} \left((H_{\text{logical}_3})_{\alpha_i\beta_i\gamma_i} + (H_{\text{logical}_4})_{\delta_i\varepsilon_i} \right)$. Next, enforce valid clock states using the following Hamiltonian, where the annotations on the left explains the function of each projector.

$$\mathbf{U} \text{ implies } \mathbf{u} \text{ to the right:} \qquad H_{\text{clock},1} = \sum_{i=1}^{N-1} \sum_{v \in \{a,d\}} |1,v\rangle\langle 1,v|_{\varepsilon_i\alpha_{i+1}} \tag{10a}$$
$$\mathbf{d} \text{ implies } \mathbf{d} \text{ to the left:} \qquad + \sum_{i=1}^{N-1} |1,d\rangle\langle 1,d|_{\beta_i\alpha_{i+1}} \tag{10b}$$
$$\mathbf{d} \text{ implies } \mathbf{D} \text{ to the left:} \qquad + \sum_{i=1}^{N-1} \sum_{v \in \{u,a_1,a_2\}} |v,1\rangle\langle v,1|_{\varepsilon_i\gamma_{i+1}} \tag{10c}$$
$$\mathbf{u}, \mathbf{a} \text{ implies } \mathbf{U} \text{ to the right:} \qquad + \sum_{i=1}^{N} \sum_{v \in \{a_1,a_2,d\}} |1,v\rangle\langle 1,v|_{\beta_i\delta_i} \tag{10d}$$
$$\mathbf{U} \text{ or } \mathbf{A}_1 \text{ is last:} \qquad + |a_2\rangle\langle a_2|_{\delta_N} + |d\rangle\langle d|_{\delta_N} \tag{10e}$$

We then have $\mathcal{N}(H_{\text{logical}} + H_{\text{clock},1}) = \text{Span}(\{|C_{i,j}\rangle \mid i \in [N], j \in [4]\} \cup \{|E_{i,j}\rangle \mid i \in \{2,\ldots,N\}, j \in [4]\})$, where the $|E_{i,j}\rangle$ are similar to the $|C_{i,j}\rangle$ but with $|\mathbf{dA}_1, \mathbf{aU}\rangle$, $|\mathbf{dA}_2, \mathbf{aU}\rangle$, $|\mathbf{dD}, \mathbf{uU}\rangle$ in the middle. The $|E_{i,j}\rangle$ are still in the nullspace because $H_{\text{clock},1}$ cannot yet

penalize all logical states that are not valid clock states. Next, we construct $H_{\text{clock},2} = \sum_{i=1}^{N} H_{\text{clock},2,i}$ to enforce the superposition $|C_{i,1}\rangle + \cdots + |C_{i,4}\rangle$, and also penalize the $|E_{i,j}\rangle$ using a similar argument to the *clairvoyance lemma* [2]. The idea is that the transition terms of (11) "evolve" the states $|E_{i,j}\rangle$ to states that are penalized by Equation (10). $H_{\text{clock},2,i}$ is given by

$$(C_{i,1} \leftrightarrow C_{i,2})\ |\mathbf{A}_2, \mathbf{u}\rangle \leftrightarrow |\mathbf{D}, \mathbf{a}\rangle: \quad \begin{cases} (|x,0\rangle - |x,1\rangle)(\langle x,0| - \langle x,1|)_{\alpha_1\beta_1}, & i = 1 \\ (|a_2,0\rangle - |d,1\rangle)(\langle a_2,0| - \langle d,1|)_{\delta_{i-1}\beta_i}, & i > 1 \end{cases} \tag{11a}$$

$$(C_{i,2} \leftrightarrow C_{i,3})\ |\mathbf{a}, \mathbf{U}\rangle \leftrightarrow |\mathbf{d}, \mathbf{U}\rangle: \quad + (|a,1\rangle - |d,1\rangle)(\langle a,1| - \langle d,1|)_{\alpha_i\varepsilon_i} \tag{11b}$$

$$(C_{i,3} \leftrightarrow C_{i,4})\ |\mathbf{d}, \mathbf{U}\rangle \leftrightarrow |\mathbf{d}, \mathbf{A}_1\rangle: \quad + (\sqrt{2}|1,u\rangle - |1,a_1\rangle)(\sqrt{2}\langle 1,u| - \langle 1,a_1|)_{\gamma_i\delta_i}. \tag{11c}$$

Finally, we set $H_{\text{clock}} = H_{\text{logical}} + H_{\text{clock},1} + H_{\text{clock},2}$ and prove $\mathcal{N}(H_{\text{clock}}) = \mathcal{C}$ and $\gamma(H_{\text{clock}}) = \Omega(1)$ in the full version [40]. ◀

The transition and selection operators are then defined as follows, where we give two variants of the $C_{\sim i}$ projectors, one acting on a 5-qudit and the other on a qubit.

$$h_{i,i+1}(U_Z) = \frac{1}{2}\left(\mathbb{I}_Z \otimes |a_1\rangle\langle a_1|_{\delta_i} + \mathbb{I}_Z \otimes |a_2\rangle\langle a_2|_{\delta_i} - U_Z^\dagger \otimes |a_1\rangle\langle a_2|_{\delta_i} - U_Z \otimes |a_2\rangle\langle a_1|_{\delta_i}\right),$$
$$C_{\leq i}^{(5)} = |u\rangle\langle u|_{\delta_i}, \quad C_{\leq i}^{(2)} = |1\rangle\langle 1|_{\varepsilon_i}, \quad C_{\geq i}^{(5)} = |d\rangle\langle d|_{\alpha_i}, \quad C_{\geq i}^{(2)} = |1\rangle\langle 1|_{\gamma_i} \tag{12}$$

Thus, the Hamiltonians from Theorem 3.1 can all be implemented as a (2×5)-projector.

3.4 QMA$_1$-completeness

Our main contribution is certainly the QMA$_1$-hardness (Theorem 1.1), but we still need to discuss containment in QMA$_1$ briefly.

▶ **Lemma 3.6.** *The* (2×5)-QSAT *and* (3×4)-QSAT *instances constructed from* QMA$_1$*-circuits are contained in* QMA$_1$*.*

Proof. To evaluate $(d_A \times d_B)$-QSAT instances with a QMA$_1$-verifier, we embed each qudit into $\lceil \log d \rceil$ qubits, and add additional diagonal projectors to reduce the local dimension as necessary. The lemma then follows from the containment of QSAT in QMA$_1$ [24], which requires projectors of a specific form. Besides (11c), all projectors used in Section 3.3 have entries in $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$. Measurements with respect to such projectors are implemented with [23]. Recall the projector from (11c), $\Pi = \left(\sqrt{\frac{2}{3}}|1,u\rangle - \sqrt{\frac{1}{3}}|1,a_1\rangle\right)\left(\sqrt{\frac{2}{3}}\langle 1,u| - \sqrt{\frac{1}{3}}\langle 1,a_1|\right)_{\gamma_i\delta_i}$. Under the qubit embedding, there exists a permutation P such that

$$P\Pi P^\dagger = \left(\sqrt{\frac{2}{3}}|0000\rangle - \sqrt{\frac{1}{3}}|0001\rangle\right)\left(\sqrt{\frac{2}{3}}\langle 0000| - \sqrt{\frac{1}{3}}\langle 0001|\right). \tag{13}$$

A measurement algorithm with perfect completeness is given in [24, Appendix A] for a 3-local projector of analogous structure, which can easily be extend to larger projectors. ◀

▶ **Theorem 1.1.** (2×5)-QSAT *is* QMA$_1$*-complete with soundness* $\Omega(1/N^2)$*.*

Proof. Follows from Theorem 3.1 and Lemma 3.5. ◀

▶ **Theorem 1.2.** (3×4)-QSAT *is* QMA$_1$*-complete with soundness* $\Omega(1/N^2)$*.*

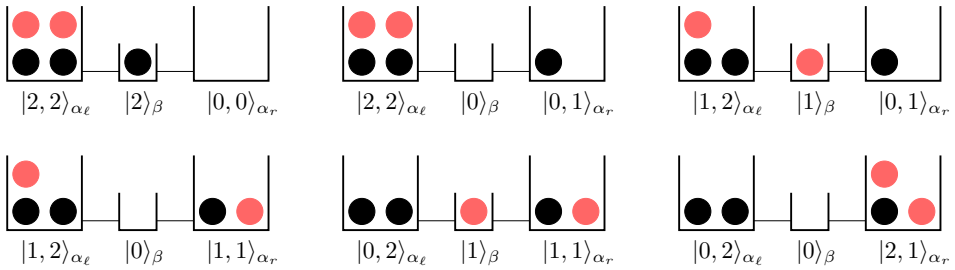


Figure 4.1 Configurations of the balls and bins for $d = 2$. Only configurations in $\mathcal{C}_1$ are depicted ($c_{1,1} = 2$ red balls and $c_{1,2} + 1 = 3$ black balls). The first two are in $\mathcal{C}_1^*$. The configurations evolve according to the transitions of (17b) and (17c). There are $d'' = 15$ possible configurations for the large bins.

Proof. We use the same clock construction operating directly in logical space, though care needs to be taken that everything is realizable with (3×4)-constraints. To implement $H_{\text{clock},1}$ using only (3×4)-terms, we have to replace "**d** implies **d** to the left" with "**D** implies **d** to the left". $H_{\text{clock},2}$ only uses (3×4)-transitions. The $C_{\sim i}$ projectors are implemented as $C^{(4)}_{\leq i} = |U\rangle\langle U|_{\delta_i}, C^{(3)}_{\leq i} = |u\rangle\langle u|_{\alpha_i}, C^{(4)}_{\geq i} = |D\rangle\langle D|_{\delta_{i-1}}, C^{(3)}_{\geq i} = |d\rangle\langle d|_{\alpha_i}$, where α_i, δ_i denote the pairs of (3×4)-qudits as depicted in Equation (9). The computational register is implemented on qutrits restricted to a 2-dimensional subspace. ◀

4 (3×d)-QSAT on a line

▶ **Theorem 1.4.** (3×d)-QSAT *on a line is* QMA$_1$*-complete with* $d = O(1)$.

To prove this theorem, we give a general construction to embed an arbitrary Hamiltonian $H = \sum_{i=1}^{n-1} H_{i,i+1}$ on n qu-d-its with d into a Hamiltonian H' on an alternating line of $n+1$ qu-d'-its ($d' = (d'')^2$) and n qutrits. The qu-d'-its are treated as two qu-d''-its of dimension $d'' \in O(d^2)$. Then each triple (d''×3×d'') logically stores one qu-d-it, which is "sent" between the outer qu-d''-its with a history-state-like construction. Conceptually, we think of this system as three bins with two kinds of balls (say *red* and *black*). The outer bins (qu-d''-its) may contain up to $B = 2d$ balls, and the middle bin only at most a single ball. A valid configuration of the bins has $B + 1$ balls, and the number of red balls is even and positive. We use transition terms to enforce a superposition of all valid configurations that have the same number of red balls. Hence, the (d''×3×d'') system has a nullspace of dimension d.

The standard basis states of the qu-d''-it are written as $|c_1, c_2\rangle$ with $c_1, c_2 \in \mathbb{Z}_{\geq 0}$ and $c_1 + c_2 \leq B := 2d$. Thus, we get $d'' = \sum_{i=0}^{B}(B+1-i) = (B+1)(B+2)/2 = (d+1)(2d+1)$. Semantically, one may think of c_1 as "number of red balls" and c_2 as "number of black balls" (see Figure 4.1). For $i \in [d]$, let $c_{i,1} = 2i, c_{i,2} = B - 2i$ and define the set of valid configurations corresponding to the state $|i\rangle \in \mathbb{C}^d$ as

$$\mathcal{C}_i =: \left\{ (l_1, l_2, m, r_1, r_2) \;\middle|\; \begin{array}{c} l_1, l_2, r_1, r_2 \in \{0, \ldots, B\} \\ m \in \{0, 1, 2\} \\ l_1 + r_1 + \delta_{m,1} = c_{i,1} \\ l_2 + r_2 + \delta_{m,2} = c_{i,2} + 1 \end{array} \right\}, \tag{14}$$

where $\delta_{x,y}$ denotes the Kronecker delta. The constraints in (14) enforce that in total there are $c_{i,1}$ red balls and $c_{i,2} + 1$ black balls (see also Figure 4.1).

We place the local terms of the original Hamiltonian into the dimensions corresponding to $\mathcal{C}_i^* \coloneqq \{(l_1, l_2, m, r_1, r_2) \in \mathcal{C}_i \mid (l_1, l_2) = c_i^* \vee (r_1, r_2) = c_i^*\}$, where $c_i^* \coloneqq (c_{i,1}, c_{i,2})$. These configurations can be identified locally, i.e., one can tell which $\mathcal{C}_i^*$ a configuration corresponds to by only looking at the left or the right bin. Note that $|\mathcal{C}_i^*| = 4$ and $\mathcal{C}_i^* \cap \mathcal{C}_j = \emptyset$ for $j \neq i$. Thus, $\langle c_i^*|_{\alpha_\ell} |\psi_j\rangle = 0$ if $j \neq i$.

A logical $|i\rangle$ is then represented by

$$|\psi_i\rangle = \sum_{x=(l_1,l_2,m,r_1,r_2)\in\mathcal{C}_i} \sqrt{w_x}|l_1, l_2\rangle_{\alpha_\ell}|m\rangle_\beta|r_1, r_2\rangle_{\alpha_r},\ w_x = \begin{cases} \frac{2}{3}\cdot\frac{1}{|\mathcal{C}_i^*|} =: w_i^*, & x \in \mathcal{C}_i^* \\ \frac{1}{3}\cdot\frac{1}{|\mathcal{C}_i\setminus\mathcal{C}_i^*|} =: \overline{w}_i, & x \notin \mathcal{C}_i^* \end{cases}, \tag{15}$$

where α_ℓ, α_r denote the qu-d''-its and β the qutrit. Let

$$V = \sum_{i=1}^{d} |\psi_i\rangle\langle i| \in \mathbb{C}^{3(d'')^2 \times d} \tag{16}$$

be the isometry that maps $|i\rangle$ to $|\psi_i\rangle$. The weights w_x in (15) ensure that the $\mathcal{C}_i^*$ always have the same amplitude ($\sqrt{1/6}$), as the $\mathcal{C}_i$ can have different sizes. The reason for having the additional configurations $\mathcal{C}_i \setminus \mathcal{C}_i^*$ is so that we can use 2-local transitions (see (17b) and (17c)) on the line to enforce a superposition between the $\mathcal{C}_i^*$ states.

Next, we construct a Hamiltonian whose nullspace is spanned by the logical states $|\psi_1\rangle, \ldots, |\psi_d\rangle$. Let $H_{\text{ball}} = (H_{\text{ball}/2})_{\alpha_\ell\beta} + (H_{\text{ball}/2})_{\alpha_r\beta}$, where α_ℓ, α_r denote the qu-d''-its and β the qutrit.

$$H_{\text{ball}/2} = P(|0,0\rangle|0\rangle) + P(|0,B\rangle|2\rangle) + P(|B,0\rangle|1\rangle) + \sum_{c_1+c_2=B, c_1 \text{ odd}} P(|c_1,c_2\rangle|2\rangle) \tag{17a}$$

$$+ \sum_{c_1>0, c_2} \big[T((c_1,c_2,0),(c_1-1,c_2,1)) + T((c_1,c_2,2),(c_1-1,c_2+1,1))\big] \tag{17b}$$

$$+ \sum_{c_1,c_2>0} \big[T((c_1,c_2,0),(c_1,c_2-1,1)) + T((c_1,c_2,1),(c_1+1,c_2-1,2))\big], \tag{17c}$$

where $P(|\psi\rangle) \coloneqq |\psi\rangle\langle\psi|$,

$$T((a_1,a_2,m_a),(b_1,b_2,m_b)) \coloneqq P(\sqrt{w_b}|a_1,a_2\rangle|m_a\rangle - \sqrt{w_a}|b_1,b_2\rangle|m_b\rangle), \tag{18a}$$

$$(w_a, w_b) = \begin{cases} (w_i^*, \overline{w}_i), & (a_1,a_2) = c_i^* \\ (\overline{w}_i, w_i^*), & (b_1,b_2) = c_i^* \\ (1,1), & \text{otherwise} \end{cases}. \tag{18b}$$

One may interpret $P(|\psi\rangle)$ as penalizing $|\psi\rangle$ and $T(\cdot)$ as a transition between the two given configurations, where the weights are chosen according to the weights in the $|\psi_i\rangle$ states.

▶ **Lemma 4.1.** $\mathcal{N}(H_{\text{ball}}) = \text{Span}\{|\psi_i\rangle \mid i \in [d]\} =: \mathcal{L}_{\text{ball}}$.

Proof. In the full version [40]. ◀

The logical states $|\psi_1\rangle, \ldots, |\psi_d\rangle$ are orthonormal and can be "identified" by projecting either qudit onto $|c_i^*\rangle$ as $\langle c_i^*|_{\alpha_r}|\psi_i\rangle = \sqrt{1/6}(|0,0\rangle|2\rangle + |0,1\rangle|0\rangle)_{\alpha_\ell\beta} =: \sqrt{1/3}|\eta\rangle$. $|\eta\rangle$ is the residual state of $|\psi_i\rangle$ after projecting onto $|c_i^*\rangle$ and is the same for all $i \in [d]$. In Figure 4.1, $|\eta\rangle_{\alpha_r\beta}$ is the superposition of the right halves of the first two configurations.

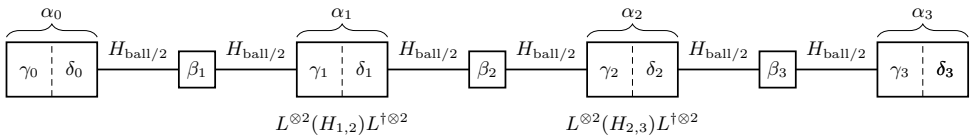


Figure 4.2 Graphical representation of H' embedding an $n = 3$ qudit Hamiltonian. The qu-d'-its $\alpha_0, \ldots, \alpha_3$ are subdivided into qu-d''-its γ_i, δ_i. H' acts trivially on γ_0, δ_3.

We define the isometry $L \coloneqq \sum_{i=1}^{d} |c_i^*\rangle\langle i| \in \mathbb{C}^{d''\times d}$ and finally the complete Hamiltonian H' on qu-d'-its $\alpha_0, \ldots, \alpha_n$ (each logically divided into two qu-d''-its γ_i and δ_i) and qubits $\beta_1, \ldots, \beta_n$ is given by $H' = H'_{\text{logical}} + H'_{\text{sim}}$ with

$$H'_{\text{logical}} = \sum_{i=1}^{n} \underbrace{\left((H_{\text{ball}/2})_{\delta_{i-1}\beta_i} + (H_{\text{ball}/2})_{\gamma_i\beta_i}\right)}_{(H_{\text{ball}})_{\delta_{i-1}\beta_i\gamma_i}} \tag{19a}$$

$$H'_{\text{sim}} = \sum_{i=1}^{n-1} (L\otimes L)(H_{i,i+1})(L\otimes L)^\dagger_{\alpha_i}, \tag{19b}$$

where H'_{logical} contains the terms of H_{ball} to enforce the logical subspace, and H'_{sim} embeds the terms of the original Hamiltonian H. Figure 4.2 gives a graphical representation of H'. Note that H' acts as identity on the first half of the first qu-d'-it (γ_0) and the second half of the last qu-d'-it (δ_n). So we can write

$$H' = \mathbb{I}_{\gamma_0} \otimes H''_{\delta_0\alpha_1\beta_1\ldots\alpha_{n-1}\beta_n\gamma_n} \otimes \mathbb{I}_{\delta_n}. \tag{20}$$

The next lemma shows that H' and H are equal inside the nullspace of H'_{logical}, up to an isometry.

▶ **Lemma 4.2.** *$T^\dagger H'' T = \frac{1}{9} H$ with $T = V^{\otimes n}$, for V and H'' as in Equations (16) and (20).*

Proof. It suffices to check equality for computational basis states $|x\rangle$ with $x \in [d]^n$. Then $T|x\rangle = |\psi_{x_1}\rangle \cdots |\psi_{x_n}\rangle =: |\psi_x\rangle$. Clearly, $H'_{\text{logical}}|\psi_x\rangle = 0$. Consider now the first summand of H'_{sim}, $(L\otimes L)(H_{1,2})(L\otimes L)^\dagger_{\alpha_i} =: M_1$. We have

$$\begin{aligned} V^{\dagger\otimes 2} M_1 |\psi_{x_1}\rangle_{\delta_0\beta_1\gamma_1} |\psi_{x_2}\rangle_{\delta_1\beta_2\gamma_2} &= \frac{1}{3} V^{\dagger\otimes 2} |\eta\rangle_{\delta_0\beta_1} \otimes L^{\otimes 2} H_{1,2} |x_1\rangle_{\varepsilon_1} |x_2\rangle_{\varepsilon_2} \otimes |\eta\rangle_{\gamma_2\beta_2} \\ &= \frac{1}{9} H_{1,2} |x_1 x_2\rangle, \end{aligned} \tag{21}$$

where $\varepsilon_1, \varepsilon_2$ denote the qudits $H_{1,2}$ acts on, and the second equality follows from the fact that $V^\dagger(|\eta\rangle_{\alpha_\ell\beta} \otimes L_{\alpha_r}) = \mathbb{I}/\sqrt{3}$. Since M_1 only acts nontrivially on the first two logical qudits, we have $T^\dagger M_1 T|x\rangle = \frac{1}{9} H|x\rangle$. Applying the same argument to the other summands of H'_{sim} yields $T^\dagger H'' T|x\rangle = \frac{1}{9} H|x\rangle$. ◀

▶ **Lemma 4.3.** *Let H be a Hamiltonian on a line of n qu-d-its with $H_{i,i+1} \succeq 0$ and $\gamma(H_{i,i+1}) \in \Omega(1)$ for all $i \in [n-1]$. There exists an efficiently computable Hamiltonian H' on an alternating chain of $n+1$ qu-d'-its and n qutrits with $d' = ((d+1)(2d+1))^2 \in O(d^4)$, such that $\lambda_{\min}(H') = 0$ iff $\lambda_{\min}(H) = 0$ and $\gamma(H') \in \Omega(\gamma(H)/\|H\|)$.*

Proof. If there exists $|\psi\rangle$ such that $H|\psi\rangle = 0$, then $H'' T|\psi\rangle = 0$ by Lemma 4.2. To prove $\gamma(H') \in \Omega(\gamma(H)/\|H\|)$, apply Corollary 2.5. We have $\mathcal{N}(H'_{\text{logical}}) = \mathcal{L}_{\text{ball}}^{\otimes n} =: S$ and $\gamma(H'_{\text{logical}}), \gamma(H'_{\text{sim}}) \in \Omega(1)$ since $H'_{\text{logical}}, H'_{\text{sim}}$ are sums of commuting Hamiltonians. Since $T^\dagger \Pi_S = T^\dagger$, it follows that $T^\dagger H'' T = \frac{1}{9} H$ is equal to $H''|_S$ up to change of basis. Hence $\gamma(H'|_S) = \frac{1}{9}\gamma(H)$ and $\|H'|_S\| = \frac{1}{9}\|H\|$. ◀

Proof of Theorem 1.4. 2-QSAT one a line of qu-d-its with $d = 11$ is QMA$_1$-complete [34]. Using Lemma 4.3, we can embed this QSAT instance into a $(3\times d')$-QSAT on a line. ◀

▶ Remark 4.4. Care needs to be taken regarding containment in QMA$_1$, since the transition terms of $H_{\text{ball}/2}$ include irrational amplitudes (see (15) and (18a)) for which the techniques of [24] (see Lemma 3.6) do not directly apply. If we allow the QMA$_1$-verifier to use gates with entries from some algebraic field extension (as in [8]), we can easily verify H'. If we are restricted to the "Clifford + T" gate set as in Definition 2.1, we can modify $H_{\text{ball}/2}$ so that the sets $\mathcal{C}_i$ are of equal size for each i. We can do this by adding additional dimensions with transitions to $|c_i^*\rangle$, which increases d' by a constant factor as $|\mathcal{C}_i| = O(d)$. Then we can set all weights in the transitions of H_{ball} to 1 so that the logical states $|\psi_i\rangle$ are just uniform superpositions over the $\mathcal{C}_i$ configurations. Lemma 4.2 then still holds, albeit with a smaller factor that depends on d.

▶ Remark 4.5 (Hamiltonian simulation). Our embedding of the Hamiltonian H into H' is related to the notion of Hamiltonian simulation [7, 13]. In a sense, our embedding is almost a *perfect simulation* [13, Definition 20], but then only the nullspace is really simulated perfectly since H'_{logical} does not commute with H'_{sim}. Our construction takes the form $H' = H'_{\text{logical}} + H'_{\text{sim}}$, such that $T^\dagger H'_{\text{sim}} T = cH$ for some constant c, where T is an isometry with $TT^\dagger = \Pi_{\mathcal{N}(H'_{\text{logical}})}$. This notion of simulation may be helpful for future quantum SAT research.

5 Hamiltonian with unique entangled ground state on a (2×4)-Line

We were only able to prove QMA$_1$-hardness of quantum SAT on a $(3\times d)$-line. This raises the question whether hardness still holds with qubits instead of qutrits. In this section, we show that it is at least possible to construct a (2×4)-QSAT instance on a line with a unique entangled null state. Therefore, the $(2\times d)$-QSAT problem on a line does not necessarily have a product state solution like 2-QSAT.

▶ **Theorem 1.5.** *Consider a line of $2n$ particles such that the i-th particle has dimension 2 for even i and 4 for odd i. There exists a Hamiltonian $H = \sum_{i=1}^{n} A_{2i-1,2i} + \sum_{i=1}^{n-1} B_{2i,2i+1} + L_{1,2} + R_{2n-1,2n}$, where A, B, L, R are 2-local projectors, such that the nullspace of H is $\mathcal{N}(H) = \text{Span}\{|\psi\rangle\}$ and $|\psi\rangle$ is entangled across all cuts.*

▶ Remark 5.1. Observe that besides the left and right boundary (projectors L, R), all (4×2)-projectors have the same form A and all (2×4)-projectors have the same form B. Therefore, H may be considered translation invariant, in a weaker sense. The Hamiltonian with a fully entangled ground state on a line of qutrits [9] also has additional projectors on the boundary.

We begin by explicitly writing the unique ground state of this Hamiltonian on line of 6 particles. The dimensions of these particles are $(4, 2, 4, 2, 4, 2)$, although for the first and second to last particle a qutrit would suffice as the the $|0\rangle/|3\rangle$ dimension is not used.

$$
\begin{aligned}
&|10|00|00\rangle && +|20|20|10\rangle\\
+&|21|00|00\rangle && +|31|20|10\rangle\\
+&|20|10|00\rangle && +|20|31|10\rangle\\
+&|31|10|00\rangle && +|31|31|10\rangle\\
+&|20|21|00\rangle && +|20|20|21\rangle\\
+&|31|21|00\rangle && +|31|20|21\rangle\\
& && +|20|31|21\rangle\\
& && +|31|31|21\rangle
\end{aligned}
\tag{22}
$$

While this state might look complex at a first glance, it can easily be understood semantically. We again think of particles as bins holding balls, but now there is only one kind of ball. A state $|c\rangle$ means that the bin holds c balls. Thus, a qu-d-it can hold at most $d-1$ balls and we have *large* bins of capacity 3 , and *small* bins of capacity 1. Initially, only the first bin contains a ball (first state in the superposition). Then we evolve according to the following rules (also in the reverse):

- If a large bin is not empty and the bin to the right is empty, we can add a ball to both bins ($|c,0\rangle \leftrightarrow |c+1,1\rangle$ for $c \in [1, d-2]$).
- If a small bin contains a ball and the large bin to the right is empty, we can move the ball from the small to the large bin ($|1,0\rangle \leftrightarrow |0,1\rangle$).

We can simplify the transitions by factoring $|*\rangle := \sqrt{1/2}(|20\rangle + |31\rangle)$. To obtain a uniform superposition, we also need to change the amplitudes of the transitions. On $2n$ particles, we obtain the following state.

$$|\phi\rangle := \sum_{i=1}^{n} \begin{pmatrix} |*\rangle^{\otimes i-1} \otimes |\,1\,0\,\rangle \otimes |\,0\,0\,\rangle^{\otimes n-i} \\ +\,|*\rangle^{\otimes i-1} \otimes |\,2\,1\,\rangle \otimes |\,0\,0\,\rangle^{\otimes n-i} \end{pmatrix} \tag{23}$$

Note, $|\phi\rangle$ has a quite similar structure to common clock constructions and can be extended to $2n$ particles. We will show that it is the unique ground state of the following Hamiltonian:

$$H = |0\rangle\langle 0|_1 + |3\rangle\langle 3|_{2n-1} + \sum_{i=1}^{n} A_{2i-1,2i} + \sum_{i=1}^{n-1} B_{2i,2i+1} \tag{24}$$

$$A = \big(|10\rangle - |21\rangle\big)\big(\langle 10| - \langle 21|\big) + \big(|20\rangle - |31\rangle\big)\big(\langle 20| - \langle 31|\big) + |30\rangle\langle 30| + |11\rangle\langle 11| \tag{25}$$

$$B = \big(|10\rangle - \sqrt{2}|01\rangle\big)\big(\langle 10| - \sqrt{2}\langle 01|\big) \tag{26}$$

▶ **Lemma 5.2.** *$|\phi\rangle$ is fully entangled, i.e. $|\phi\rangle \neq |\phi_1\rangle_A \otimes |\phi_2\rangle_B$ for all cuts A/B and $|\phi_1\rangle, |\phi_2\rangle$.*

Proof. Consider the random experiment of measuring $|\psi\rangle$ in standard basis. The outcome is denoted by the string x. Let $S \subset [2n]$ be a subset of particles. If $|\phi\rangle = |\phi_S\rangle|\phi_{\bar{S}}\rangle$, then the random variables x_S and $x_{\bar{S}}$ (substrings of x on particles S and $\bar{S} = [2n] \setminus S$, respectively) are independent. In the following, we argue that this is not the case.

Note for an odd i, $P(x_i = 3, x_{i+1} = 0) = 0$, but $P(x_i = 3)P(x_{i+1} = 0) > 0$. Hence, if there exists odd i such that $|\{i, i+1\} \cap S| = 1$, then $x_S, x_{\bar{S}}$ are not independent.

Otherwise, there exists an odd i such that $|\{i, i+2\} \cap S| = 1$. Again, $x_S, x_{\bar{S}}$ are not independent as $P(x_i = 0, x_{i+2} = 1) = 0$, but $P(x_i = 0)P(x_{i+2} = 1) > 0$. ◀

▶ **Lemma 5.3.** *$|\phi\rangle$ is the unique ground state of H.*

Proof. It is easy to verify $H|\phi\rangle = 0$. Now, assume $H|\psi\rangle = 0$. If there exists a standard basis vector $|x\rangle$ such that $\langle x|\psi\rangle \neq 0$, it corresponds to an illegal state of the ball game (terms of (23) are the legal states). Observe that the transition terms of A and B directly correspond to the allowed moves in the ball game. The illegal states that are not caught directly, are `|10|` or `|21|` not followed by all zeroes. By applying the transition rules, we can go to `|11|`, which is caught by A. Hence, $|\psi\rangle$ and $|\phi\rangle$ overlap the same standard basis vectors. The transition terms ensure the weights are such that $|\phi\rangle$ can be written as in (5.2). ◀

▶ Remark 5.4. $|\phi\rangle$ has only constant entanglement entropy, whereas [9] achieves $\Omega(\log n)$. So it remains an open problem whether logarithmic entanglement entropy can be achieved on the (2×4)-line.

References

1. D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev. Adiabatic quantum computation is equivalent to standard quantum computation. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 42–51, 2004. doi:10.1109/FOCS.2004.8.
2. Dorit Aharonov, Daniel Gottesman, Sandy Irani, and Julia Kempe. The power of quantum systems on a line. *Communications in Mathematical Physics*, 287(1):41–65, April 2009.
3. Dorit Aharonov and Leo Zhou. Hamiltonian Sparsification and Gap-Simulation. In Avrim Blum, editor, *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:21, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2019.2.
4. Itai Arad, Miklos Santha, Aarthi Sundaram, and Shengyu Zhang. Linear Time Algorithm for Quantum 2SAT. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:14, Dagstuhl, Germany, 2016. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2016.15.
5. Bengt Aspvall, Michael F. Plass, and Robert Endre Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8(3):121–123, 1979. doi:10.1016/0020-0190(79)90002-4.
6. Johannes Bausch, Toby Cubitt, and Maris Ozols. The complexity of translationally invariant spin chains with low local dimension. *Annales Henri Poincaré*, 18(11):3449–3513, November 2017. doi:10.1007/s00023-017-0609-7.
7. S. Bravyi and M. Hastings. On complexity of the quantum Ising model. *Communications in Mathematical Physics*, 349(1):1–45, 2017. doi:10.1007/s00220-016-2787-4.
8. Sergey Bravyi. Efficient algorithm for a quantum analogue of 2-sat, 2006. arXiv:quant-ph/0602108.
9. Sergey Bravyi, Libor Caha, Ramis Movassagh, Daniel Nagaj, and Peter W. Shor. Criticality without frustration for quantum spin-1 chains. *Phys. Rev. Lett.*, 109:207202, November 2012. doi:10.1103/PhysRevLett.109.207202.
10. Sergey Bravyi, David P. DiVincenzo, and Daniel Loss. Schrieffer–Wolff transformation for quantum many-body systems. *Annals of Physics*, 326(10):2793–2826, 2011. doi:10.1016/j.aop.2011.06.004.
11. Jianxin Chen, Xie Chen, Runyao Duan, Zhengfeng Ji, and Bei Zeng. No-go theorem for one-way quantum computing on naturally occurring two-level systems. *Phys. Rev. A*, 83:050301, May 2011. doi:10.1103/PhysRevA.83.050301.
12. Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA, May 1971. Association for Computing Machinery. doi:10.1145/800157.805047.
13. T. S. Cubitt, A. Montanaro, and S. Piddock. Universal quantum Hamiltonians. *National Academy of Sciences*, 115(38):9497–9502, 2018. doi:10.1073/pnas.1804949115.
14. Toby Cubitt and Ashley Montanaro. Complexity Classification of Local Hamiltonian Problems. *SIAM Journal on Computing*, 45(2):268–316, January 2016. doi:10.1137/140998287.
15. M. Davis and H. Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7(3):201, 1960.
16. Niel de Beaudrap and Sevag Gharibian. A Linear Time Algorithm for Quantum 2-SAT. In Ran Raz, editor, *31st Conference on Computational Complexity (CCC 2016)*, volume 50 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:21, Dagstuhl, Germany, 2016. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2016.27.
17. Lior Eldar and Oded Regev. Quantum sat for a qutrit-cinquit pair is qma1-complete. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming*, pages 881–892, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. doi:10.1007/978-3-540-70575-8_72.

18 S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM Journal on Computing*, 5(4):691–703, 1976. doi:10.1137/0205048.

19 Richard P. Feynman. Quantum mechanical computers. *Foundations of Physics*, 16(6):507–531, June 1986. doi:10.1007/BF01886518.

20 M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, February 1976. doi:10.1016/0304-3975(76)90059-1.

21 Sevag Gharibian. Strong np-hardness of the quantum separability problem. *Quantum Info. Comput.*, 10(3):343–360, March 2010. doi:10.26421/QIC10.3-4-11.

22 Sevag Gharibian, Stephen Piddock, and Justin Yirka. Oracle Complexity Classes and Local Measurements on Physical Hamiltonians. In Christophe Paul and Markus Bläser, editors, *37th International Symposium on Theoretical Aspects of Computer Science (STACS 2020)*, volume 154 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:37, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.STACS.2020.20.

23 Brett Giles and Peter Selinger. Exact synthesis of multiqubit clifford+t circuits. *Phys. Rev. A*, 87:032332, March 2013. doi:10.1103/PhysRevA.87.032332.

24 David Gosset and Daniel Nagaj. Quantum 3-sat is qma1-complete. *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, October 2013. doi:10.1109/focs.2013.86.

25 Leonid Gurvits. Classical deterministic complexity of edmonds' problem and quantum entanglement. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 10–19, New York, NY, USA, 2003. Association for Computing Machinery. doi:10.1145/780542.780545.

26 S. Hallgren, D. Nagaj, and S. Narayanaswami. The Local Hamiltonian problem on a line with eight states is QMA-complete. *Quantum Information & Computation*, 13(9&10):0721–0750, 2013. doi:10.26421/QIC13.9-10-1.

27 Lawrence M. Ioannou. Computational complexity of the quantum separability problem. *Quantum Info. Comput.*, 7(4):335–370, May 2007. doi:10.26421/QIC7.4-5.

28 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*, pages 85–103, Boston, MA, 1972. Springer US. doi:10.1007/978-1-4684-2001-2_9.

29 Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097, 2006. doi:10.1137/S0097539704445226.

30 A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, USA, 2002.

31 M. R. Krom. The Decision Problem for a Class of First-Order Formulas in Which all Disjunctions are Binary. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 13:15–20, 1967.

32 Zeph Landau, Umesh Vazirani, and Thomas Vidick. A polynomial time algorithm for the ground state of one-dimensional gapped local Hamiltonians. *Nature Physics*, 11(7):566–569, July 2015. doi:10.1038/nphys3345.

33 L. Levin. Universal search problems. *Problems of Information Transmission*, 9(3):265–266, 1973.

34 Daniel Nagaj. Local hamiltonians in quantum computation, 2008. arXiv:0808.2117.

35 Daniel Nagaj and Shay Mozes. New construction for a QMA complete three-local Hamiltonian. *Journal of Mathematical Physics*, 48(7):072104, July 2007. doi:10.1063/1.2748377.

36 C. H. Papadimitriou. On selecting a satisfying truth assignment. In *[1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science*, pages 163–169, 1991. doi:10.1109/SFCS.1991.185365.

37 Asher Peres. Separability criterion for density matrices. *Phys. Rev. Lett.*, 77:1413–1415, August 1996. doi:10.1103/PhysRevLett.77.1413.

38 W. V. Quine. On cores and prime implicants of truth functions. *The American Mathematical Monthly*, 66(5):755–760, 1959.

39 Dorian Rudolph. Towards a universal gateset for QMA_1, 2024. arXiv:2411.02681.

40 Dorian Rudolph, Sevag Gharibian, and Daniel Nagaj. Quantum 2-SAT on low dimensional systems is QMA_1-complete: Direct embeddings and black-box simulation, 2024. arXiv:2401.02368, doi:10.48550/arXiv.2401.02368.

41 Norbert Schuch, Ignacio Cirac, and Frank Verstraete. Computational difficulty of finding matrix product ground states. *Phys. Rev. Lett.*, 100:250501, June 2008. doi:10.1103/PhysRevLett.100.250501.

42 James D. Watson. Detailed analysis of circuit-to-hamiltonian mappings, 2019. arXiv:1910.01481.

43 Stathis Zachos and Martin Furer. Probabilistic quantifiers vs. distrustful adversaries. In Kesav V. Nori, editor, *Foundations of Software Technology and Theoretical Computer Science*, pages 443–455, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.

Eulerian Orientations and Hadamard Codes: A Novel Connection via Counting

Shuai Shao
School of Computer Science and Technology & Hefei National Laboratory,
University of Science and Technology of China, Hefei, China

Zhuxiao Tang
School of the Gifted Young, University of Science and Technology of China, Hefei, China

Abstract

We discover a novel connection between two classical mathematical notions, Eulerian orientations and Hadamard codes by studying the counting problem of Eulerian orientations (#EO) with local constraint functions imposed on vertices. We present two special classes of constraint functions and a chain reaction algorithm, and show that the #EO problem defined by each class alone is polynomial-time solvable by the algorithm. These tractable classes of functions are defined inductively, and quite remarkably the base level of these classes is characterized perfectly by the well-known Hadamard code. Thus, we establish a novel connection between counting Eulerian orientations and coding theory. We also prove a #P-hardness result for the #EO problem when constraint functions from the two tractable classes appear together.

2012 ACM Subject Classification Mathematics of computing → Graph theory

Keywords and phrases Eulerian orientations, Hadamard codes, Counting problems, Tractable classes

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.86

Related Version *Full Version*: https://arxiv.org/abs/2411.02612.

Funding Supported by the Innovation Program for Quantum Science and Technology, 2021ZD0302901.

Acknowledgements The first author wants to thank Jin-Yi Cai and Zhiguo Fu for many valuable discussions. The authors want to thank anonymous reviewers for their helpful comments.

1 Introduction

The notion of Eulerian orientations arises from the historically notable *Seven Bridges of Königsberg* problem, whose solution by Euler in 1736 [14] is considered to be the first result of graph theory. Given an undirected graph G, an Eulerian orientation of G is an assignment of a direction to each edge of G such that at each vertex v, the number of incoming edges is equal to the number of outgoing edges. A connected graph has an Eulerian orientation, called an Eulerian graph if and only if every vertex has even degree. This is the well-known Euler's theorem, whose first complete proof was given back in the 1800s [3]. In terms of computational complexity, Euler's theorem implies that the decision problem of Eulerian orientations of a graph is polynomial-time solvable (tractable). While for the counting problem, Mihail and Winkler showed that counting the number of Eulerian orientations of an undirected graph is #P-complete in 1996 [18], more than one hundred years later after Euler's theorem. However, the counting problem becomes tractable when certain particular restrictions are imposed on edges. An intriguing example of such tractable problems comes from computing the partition function of the six-vertex model [19, 22, 20], one of the most intensively studied models in statistical physics. In this example, the graphs are 4 regular graphs and on each vertex exactly one edge incident to it is restricted to take the direction coming into the vertex. Then counting the number of Eulerian orientations obeying restrictions on these edges is proved to be tractable [7].


© Shuai Shao and Zhuxiao Tang;
licensed under Creative Commons License CC-BY 4.0



In this paper, we further study the counting restricted Eulerian orientations (#EO) problem, defined by constraint functions placed at each vertex that represent restrictions on edges incident to the vertex. We consider for which class of constraint functions, the #EO problem is tractable. Before we formally define the problem and describe our results, we first take a detour to another classical notion in coding theory, the Hadamard code. The Hadamard code is an error-correcting code used for error detection and correction when transmitting messages over very noisy or unreliable channels. Because of its plentiful mathematical properties, the Hadamard code is not only used in coding and communication, but also intensely studied in various areas of mathematics and theoretical computer science. However, it is not known how the Hadamard code can be related to Eulerian orientations, and particularly, how it can be used to carve out tractable classes for the #EO problem. Quite surprisingly, in this paper, we establish a novel connection between these two well-studied mathematical notions. We present new tractable classes for the #EO problem and a corresponding algorithm based on a chain reaction. These classes are defined inductively and the base levels of them, defined as kernels are characterized perfectly by the Hadamard code (or more precisely, the balanced Hadamard code).

Now we formally define the #EO problem. A 0-1 valued constraint function (or a signature) f of arity n is a map $\mathbb{Z}_2^n \to \{0,1\}$. The support of f is $\mathscr{S}(f) = \{\alpha \in \mathbb{Z}_2^n | f(\alpha) = 1\}$. A signature f of arity $2n$ is an *Eulerian orientation* (EO) signature if for every $\alpha \in \mathscr{S}(f)$, $\mathrm{wt}(\alpha) = n$. The problem #EO($\mathcal{F}$) specified by a set $\mathcal{F}$ of EO signatures is defined as follows. The input is a graph G where each vertex v of G is associated with some function f_v from $\mathcal{F}$. The incident edges to v are totally ordered and correspond to input variables to f_v. Each edge has two ends, and an orientation of the edge is denoted by assigning 0 to the head and 1 to the tail. Thus, locally at every vertex v, a 0 represents an incoming edge and a 1 represents an outgoing edge. An Eulerian orientation corresponds to an assignment to each edge (01 or 10) where the numbers of 0's and 1's at each v are equal. Then, the local constraint function f_v takes value 1 if the restriction imposed by f_v are satisfied by the local assignment on edges incident to v. Otherwise, $f_v = 0$. Since every f_v is an EO signature, it takes value 1 only if the numbers of input 0's and 1's are equal. Thus, only Eulerian orientations can contribute value 1. An Eulerian orientation contributes value 1 if $f_v = 1$ for every vertex v. The #EO problem outputs the number of Eulerian orientations contributing value 1.

▶ **Example 1.** Let $\mathcal{F} = \{g_2, g_4, \ldots g_{2n}, \ldots\}$, where $\mathscr{S}(g_{2n}) = \{\alpha \in \mathbb{Z}_2^{2n} \mid \mathrm{wt}(\alpha) = n\}$. Then #EO($\mathcal{F}$) counts the total number of all Eulerian orientations of the input graph, which is #P-hard.

▶ **Example 2.** Let f_2 be a signature of arity 4 with support $\mathscr{S}(f_2) = \{1100, 1010, 1001\}$. Then, #EO($\{f_2\}$) computes the partition function of a six-vertex model which is tractable. The reason for naming this function f_2 will be explained in Section 4.

The #EO problem has an intrinsic significance in counting problems. In [5], the framework of #EO problems is formally introduced and is showed to be expressive enough to encompass all Boolean counting constraint satisfaction problems (#CSP) [12, 13, 4, 10] with arbitrary constraint functions that are not necessarily supported on half weighted inputs, although the #EO problem itself requires all constraint functions to be supported on half weighted inputs. Also, the #EO problem encompasses many natural problems from statistical physics and combinatorics, such as the partition function of the six-vertex model and the evaluation of the Tutte polynomial at the point $(3, 3)$ [15]. Cai, Fu and Xia proved a complexity classification of the partition function of the six-vertex model on general 4-regular graphs [7].

Besides these interesting concrete problems expressible as #EO problems, the study of the #EO framework plays a crucial role in a broader picture, the complexity classification program for a much more expressive counting framework, the Holant problem. Significant progress has been made in the complexity classification of Holant problems [9, 8, 1, 11, 2, 16, 6], and a full complexity dichotomy for all real-valued Holant problems is achieved [21]. This dichotomy is crucially based on a complexity dichotomy for the #EO problem [5] with complex-valued signatures assuming a physical symmetry called *arrow reversal symmetry* (ARS) since under a suitable holographic transformation, the ARS property corresponds to precisely real-valued signatures. In this paper, we consider the 0-1 valued #EO problem without assuming ARS, which may serve as a building block for the ultimate classification for all complex-valued Holant problems. For this problem, we give new tractable classes that are not covered by all the previously known tractable classes for Holant problems.

We use $a(f)$ to denote the arity of f, and $A(f) = \{1, 2, ..., a(f)\}$ to denote the indices of variables of f. We use δ_1 and δ_0 to denote the unary signatures where $\mathscr{S}(\delta_1) = \{1\}$ and $\mathscr{S}(\delta_0) = \{0\}$ respectively. Pinning the i-th variable of f to 0 gives the signature $f(x_1, ..., x_{i-1}, 0, x_{i+1}, ..., x_n)$ of arity $n-1$, denoted by f_i^0. Similarly, we define f_i^1. Let f and g be two signatures of support size n and m respectively. Suppose $\mathscr{S}(f) = \{\alpha_1, \alpha_2, ..., \alpha_n\}$ and $\mathscr{S}(g) = \{\beta_1, \beta_2, ..., \beta_m\}$. Then the tensor product $f \otimes g$ of f and g is the signature where $\mathscr{S}(f \otimes g) = \{\alpha_i\beta_j | 1 \le i \le n, 1 \le j \le m\}$. Note that when we take a tensor product of the functions, the variables need not be in the same order of the product. A nonzero signature g is a factor of f, denoted by $g \mid f$, if there is a signature h such that $f = g \otimes h$ or there is a constant λ such that $f = \lambda \cdot g$. Otherwise, g is not factor of f, denoted by $g \nmid f$. In the following, we first state the definition of affine signatures, which are known to be tractable for the #EO problem. Then we give the new tractable classes.

▶ **Definition 3** (Affine). *A signature $f(x_1, x_2, ..., x_n)$ of arity n is affine, denoted by $f \in \mathscr{A}$ if $\mathscr{S}(f) = \{x \in \mathbb{Z}_2^n | AX = 0\}$, where $X = (x_1, x_2, ..., x_n, 1)$ and A is a matrix over $\mathbb{Z}_2$. Note that $f \in \mathscr{A}$ when $\mathscr{S}(f) = \emptyset$, is called a zero signature.*

▶ **Definition 4** (δ_1-affine and δ_0-affine). *An EO signature f is δ_1-affine, denoted by $f \in \mathscr{D}_1$ if $f = \delta_1 \otimes g$ for some g where for every $i \in A(g)$, $g_i^0 \in \mathscr{A} \cup \mathscr{D}_1$. Symmetrically, an EO signature f is δ_0-affine, denoted by $f \in \mathscr{D}_0$ if $f = \delta_0 \otimes g$ for some g where for every $i \in A(g)$, $g_i^1 \in \mathscr{A} \cup \mathscr{D}_0$.*

For example, the 4-ary signature f in Example 2 is an δ_1-affine signature. Note that $\mathscr{D}_1$ (or $\mathscr{D}_0$) does not encompass $\mathscr{A}$, since an affine signature may not have a δ_1 (or δ_0) factor. Thus, our tractable classes are $\mathscr{A} \cup \mathscr{D}_1$ and $\mathscr{A} \cup \mathscr{D}_0$, rather than $\mathscr{D}_1$ and $\mathscr{D}_0$.

▶ **Theorem 5.** *#EO($\mathscr{A} \cup \mathscr{D}_1$) and #EO($\mathscr{A} \cup \mathscr{D}_0$) are tractable.*

The tractability result is established via a chain reaction algorithm, in which the presence of a δ_1 or δ_0 signature plays a role similar to a neutron in a nuclear chain reaction which initializes the reaction and generates a new neutron (a δ_1 or δ_0 signature respectively) by propagation which makes the chain reaction continue. Eventually, the chain reaction terminates when a stable state (a tractable instance of #EO($\mathscr{A}$)) is reached.

Although the reaction algorithm works for δ_1-affine signatures or δ_0-affine signatures alone, it does not work when both of them appear. This is because when both δ_1-affine signatures and δ_0-affine signatures appear, by connecting δ_1 and δ_0 using $\neq$, both of them are killed by each other and no new δ_1 or δ_0 are generated. Thus, the chain reaction is unsustainable. This is somewhat similar to the phenomenon of electron–positron annihilation in nuclear physics. In fact, we can show that the #EO problem is #P-hard when they both appear.

▶ **Theorem 6.** $\#\mathrm{EO}(\mathscr{D}_0 \cup \mathscr{D}_1)$ *is #P-hard.*

Since the classes $\mathscr{D}_1$ and $\mathscr{D}_0$ are defined inductively, it is not an easy task to get explicit expressions for them in a form similar to other known tractable classes for counting problems such as affine signatures. Note that the tractability of $\mathscr{D}_1$ or $\mathscr{D}_0$ is obtained by eventually reducing the problem into a problem where all signatures are affine signatures. Thus, we focus on δ_1-affine (or δ_0-affine) signatures from which by pinning an arbitrary variable to 0 (or 1 respectively), we always get an affine signature. In other words, these δ_1-affine (or δ_0-affine) signatures are at the first level of the inductive hierarchy. We define them as kernels.

▶ **Definition 7** (δ_1-affine and δ_0-affine kernel)**.** *An* EO *signature f is a δ_1-affine kernel, denoted by $f \in K(\mathscr{D}_1)$ if it satisfies the following two properties:*

1. *$f = \delta_1^{\otimes m} \otimes h$, for some $m \in \mathbb{Z}_+$ where h is a signature of positive arity, $\delta_1 \nmid h$ and $h \notin \mathscr{A}$;*
2. *for every $i \in A(h)$, $h_i^0 \in \mathscr{A}$.*

Symmetrically, we define δ_0-affine kernels denoted by $K(\mathscr{D}_0)$ by switching 0 *and* 1 *everywhere. Notice that $h \notin \mathscr{A}$ implies $f \notin \mathscr{A}$. Thus, δ_1-affine and δ_0-affine kernels are not affine.*

A main technical contribution of the paper is a complete characterization of δ_1-affine and δ_0-affine kernels using the Hadamard code. The Hadamard code refers to the code based on Sylvester's construction of Hadamard matrices. A Hadamard matrix H_n is a square matrix of size n such that all its entries are in $\{1, -1\}$ and $H_n H_n^T = nI_n$. Sylvester's construction gives Hadamard matrices H_{2^k} of size 2^k for all $k \geq 0$, where $H_{2^{k+1}} = \begin{bmatrix} H_{2^k} & H_{2^k} \\ H_{2^k} & -H_{2^k} \end{bmatrix}$ and $H_{2^0} = 1$. The Hadamard code $\mathscr{H}_{2^k}^1$ derived from the Hadamard matrices H_{2^k} by Sylvester's construction is a code over the binary alphabet $\{0, 1\}$ of which the 2^k codewords are rows of H_{2^k} where the entry 1 of H_{2^k} is mapped to 1 and the entry -1 is mapped to 0. Symmetrically, we can also get the Hadamard code $\mathscr{H}_{2^k}^0$ by mapping the 1 entry to 0 and -1 to 1. Note that $\mathscr{H}_{2^k}^1$ contains an all-1 codeword and $\mathscr{H}_{2^k}^0$ contains an all-0 codeword. We call the former the 1-Hadamard code and the latter the 0-Hadamard code. In standard coding theory notation for block codes, $\mathscr{H}_{2^k}^1$ and $\mathscr{H}_{2^k}^0$ are $[2^k, k, 2^{k-1}]$-code, which is a binary linear code having block length 2^k, message length k, and minimum Hamming distance $2^k/2$. By removing the all-1 codeword from $\mathscr{H}_{2^k}^1$, we get a balanced code, i.e., each codeword has Hamming weight 2^{k-1}, half of its block length. We call it the balanced 1-Hadamard code, denoted by $\mathscr{H}_{2^k}^{1\mathrm{b}}$. Symmetrically, we can get the balanced 0-Hadamard code, denoted by $\mathscr{H}_{2^k}^{0\mathrm{b}}$. A binary code $\mathscr{C}_n$ of block length n can be viewed equivalently as a 0-1 valued constraint function f of arity n by taking the support $\mathscr{S}(f) = \{\alpha \in \mathbb{Z}_2^n | f(\alpha) \neq 0\}$ to be $\mathscr{C}_n$. An m-multiple of the code $\mathscr{C}_n$ is a code of block length nm, each codeword of which is obtained from a codeword α of $\mathscr{C}_n$ by taking the concatenation of m copies of α.

▶ **Theorem 8.** *An* EO *signature f is a δ_1-affine kernel if and only if $\mathscr{S}(f)$ is an m-multiple of a balanced 1-Hadamard code $\mathscr{H}_{2^k}^{1\mathrm{b}}$ for some $k \geq 3$ and $m \geq 1$, or $\delta_0 \nmid f$ and $|\mathscr{S}(f)| = 3$.*

A symmetric statement holds for the δ_0-affine kernel by switching 0 *and* 1 *everywhere.*

Due to the existence of newly discovered non-trivial tractable classes corresponding to the Hadamard code in the #EO problem without assuming ARS, the complexity classification for the complex-valued Holant problem is much more challenging. It's hard to imagine how many more tractable classes with remarkable properties are yet to be discovered. The results in this paper are a tiny step towards an ambitious goal. Very recently, independent of this work, Meng, Wang and Xia discover new larger tractable classes for the #EO problem [17].

In addition, we note that the presented result in this paper is not the first time that a tractable class for counting problems is found to be related to coding theory. Even in the classification of real-valued Holant problems, a tractable family called local affine [11] was already discovered using the well-known Hamming code. Are there other interesting tractable classes for counting problems that can be carved out with the help of coding theory? In the other direction, it is also worth pursuing whether these new tractability results for counting problems can be applied back to coding theory? We leave the questions in both directions for further study.

The paper is organized as follows. In Section 2, we give some definitions and notations. In Section 3, we give a chain reaction algorithm which establishes the tractability result (Theorem 5) for δ_1-affine and δ_0-affine signatures. In Section 4, we prove the characterization theorem (Theorem 8) for δ_1-affine and δ_0-affine kernels. In Section 5, we prove the hardness result Theorem 6. In fact, the proof of the hardness result highly depends on the characterization of δ_1-affine and δ_0-affine kernels. Omitted proofs can be founded in the full version.

2 Preliminaries

A 0-1 valued signature is determined by its support $\mathscr{S}(f)$. In the paper, we use f and $\mathscr{S}(f)$ interchangeably. We can also view $\mathscr{S}(f)$ as a binary code of block length $a(f)$ where each $\alpha \in \mathscr{S}(f)$ is a codeword. We say $\mathscr{S}(f)$ (or equivalently f) has a constant Hamming weight ℓ if $\mathrm{wt}(\alpha) = \ell$ for every $\alpha \in \mathscr{S}(f)$. For simplicity, we say f is ℓ-weighted or constant-weighted in general. The following fact can be easily checked. Any factor of a constant-weighted signature is still constant-weighted. If a vertex v in a graph is labeled by a signature $f = g \otimes h$, we can replace the vertex v by two vertices v_1 and v_2 and label v_1 with the factor g and v_2 with h, respectively. The incident edges of v become incident edges of v_1 and v_2 respectively according to the partition of variables of f in the tensor product of g and h. This does not affect the evaluation of the instance. We represent f or $\mathscr{S}(f)$ by a matrix which lists all vectors in $\mathscr{S}(f)$ by its rows. All matrices that are equal up to a permutation of rows and columns represent the same signature. Normally, the columns indexing variables are ordered from the smallest index to the largest, and are omitted for convenience when the context is clear. For example, the signature $f = \{1100, 1010, 1001\}$ can be written as $\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$.

For $i \in A(f)$, we use $\#_i^0(f)$ and $\#_i^1(f)$ to denote the number of 0s and 1s respectively in the i-th column of $\mathscr{S}(f)$. The complement of a signature f, denoted by $\overline{f}$ is the signature with the support $\mathscr{S}(\overline{f}) = \{\alpha \in \mathbb{Z}_2^n | \overline{\alpha} \in \mathscr{S}(f)\}$ where $\overline{\alpha}$ is the standard complement of α, i.e., $\overline{\alpha} \oplus \alpha = \vec{0}^n$. Let $\widehat{f}(x_1, x_2, ..., x_n) = x_1 x_2 ... x_n \oplus f(x_1, x_2, ..., x_n)$. Then, $\mathscr{S}(\widehat{f}) = \{\vec{1}^n\} \Delta \mathscr{S}(f)$. Similarly, let $\breve{f}(x_1, x_2, ..., x_n) = (1 \oplus x_1)(1 \oplus x_2)...(1 \oplus x_n) \oplus f(x_1, x_2, ..., x_n)$. Then $\mathscr{S}(\breve{f}) = \{\vec{0}^n\} \Delta \mathscr{S}(f)$. We use $\neq$ to denote the binary disequality signature $\{01, 10\}$. Connecting δ_1 with the i-th variable of f using $\neq$ is equivalent to pinning the i-th variable of f to 0, which gives the signature f_i^0. Extracting the i-th variable of f to 0 gives the signature $(x_i \oplus 1) f(x_1, x_2, ..., x_n)$ of arity n, denoted by $f^{x_i=0}$. Clearly $f^{x_i=0} = \delta_0 \otimes f_i^0$. Similarly we define f_i^1 and $f^{x_i=1}$. We use f_{ij}^{ab} to denote the signature $(f_i^a)_j^b = (f_j^b)_i^a$ for distinct $i, j \in A(f)$ and any $a, b \in \mathbb{Z}_2$. The signature $f^{x_i \neq x_j}$ is defined to be $f_{ij}^{01} + f_{ij}^{10}$ which can be obtained by connecting the i-th and j-th variables of f using $\neq$. For a signature f, we define its m-multiple, denoted by $f_{\times m}$ to be the signature $\mathscr{S}(f_{\times m}) = \{\vec{\alpha}^m \mid \alpha \in \mathscr{S}(f)\}$ where $\vec{\alpha}^m \in \mathbb{Z}_2^{nm}$ is obtained from α by taking the concatenation of m copies of α.

Now we give some basic properties of affine signatures. Equivalently, f is affine if and only if it satisfies the property that for any $\alpha, \beta, \gamma \in \mathscr{S}(f)$, $\alpha \oplus \beta \oplus \gamma \in \mathscr{S}(f)$. One can also check that affine signatures are closed under pinning, extracting, tensor product and factoring. Also, $f \in \mathscr{A}$ if and only if $f_{\times m} \in \mathscr{A}$ for any $m \in \mathbb{Z}_+$.

▶ **Lemma 9.** *If a signature $f \in \mathscr{A}$ is constant-weighted and $\delta_0, \delta_1 \nmid f$, then $\#_i^0(f) = \#_i^1(f)$ for any $i \in A(f)$. Moreover, f is an* EO *signature.*

▶ **Definition 10** (Pairwise opposite)**.** *Let f be a signature of arity $2n$. We say f is pairwise opposite if we can partition the $2n$ variables into n pairs such that in $\mathscr{S}(f)$, two variables of each pair always take opposite values.*

▶ **Lemma 11** (Lemma 5.7 in [5])**.** *If f is an affine* EO signature, then f is pairwise opposite.

3 A chain reaction algorithm

In the section, we show $\#\mathrm{EO}(\mathscr{D}_1 \cup \mathscr{A})$ and $\#\mathrm{EO}(\mathscr{D}_0 \cup \mathscr{A})$ are tractable by giving a chain reaction algorithm. We only consider $\#\mathrm{EO}(\mathscr{D}_1 \cup \mathscr{A})$, since the other case is symmetric. Given an instance Ω of $\#\mathrm{EO}(\mathscr{D}_1 \cup \mathscr{A})$, we may assume that there is at least one vertex labelled by a δ_1-affine signature f. Otherwise, all signatures in the instance are affine. Such an instance of $\#\mathrm{EO}(\mathscr{A})$ is tractable.

Since f is δ_1-affine, consider a factor δ_1 of f. This δ_1 is connected to another variable of some signature (which may be f itself) using $\neq_2$. Then, the connected variable is pinned to 0. A key property that ensures our algorithm work is that after connecting δ_1 to another variable no matter whether it is a variable of f itself, an affine signature, or another δ_1-affine signature, either the resulting signature is affine, or we can realize another δ_1. In the later case, we call it a propagation step. After a propagation step, the new realized δ_1 can be used to pin another variable to 0. Thus, we get a chain reaction, and it terminates when an instance of $\#\mathrm{EO}(\mathscr{A})$ is reached. Note that, after each propagation step, the number of edges (total variables) in the instance is reduced by 2. Thus, the chain reaction terminates after at most $|E|/2$ many steps where E is the edge set of the underlying graph in Ω.

▶ **Lemma 12.** *Suppose that $f = \delta_1 \otimes h$ is δ_1-affine and $g \in \mathscr{D}_1 \cup \mathscr{A}$. For any variable x_i of g, by connecting it with δ_1 of f using $\neq_2$, we can get a zero signature or a signature $h \otimes \delta_1 \otimes g_{ij}^{01}$ where $g_{ij}^{01} \in \mathscr{D}_1 \cup \mathscr{A}$ for some $j \in A(g) \backslash \{i\}$.*

Proof. If $g \in \mathscr{A}$, then g is pairwise opposite by Lemma 11. Suppose x_j is opposite to x_i. Then pinning x_i to 0 makes x_j be a δ_1 factor, i.e., $g_i^0 = \delta_1(x_j) \otimes g_{ij}^{01}$, where $g_{ij}^{01} \in \mathscr{A}$. Otherwise $g \in \mathscr{D}_1 \backslash \mathscr{A}$. Suppose $g = \delta_1(x_j) \otimes g'$ for some variable x_j. If $x_i = x_j$, then by connecting the δ_1 factor of f and x_j of g using $\neq$, we get a zero signature. Now assume $x_i \neq x_j$. Then $(g')_i^0 \in \mathscr{A} \cup \mathscr{D}_1$ since $g \in \mathscr{D}_1$. In this case, we get $h \otimes \delta_1(x_j) \otimes g_{ij}^{01}$, where $g_{ij}^{01} = (g')_i^0 \in \mathscr{A} \cup \mathscr{D}_1$. ◀

▶ **Theorem 13.** *$\#\mathrm{EO}(\mathscr{A} \cup \mathscr{D}_1)$ and $\#\mathrm{EO}(\mathscr{A} \cup \mathscr{D}_0)$ are tractable.*

Proof. We only prove $\#\mathrm{EO}(\mathscr{A} \cup \mathscr{D}_1)$ is tractable, the other case is symmetric.

Consider an instance Ω of $\#\mathrm{EO}(\mathscr{A} \cup \mathscr{D}_1)$. If all signatures in Ω are affine, then we are done. Otherwise, there is a vertex u in Ω that is labeled by a δ_1-affine signature $f = \delta_1 \otimes h$. Without loss of generality, we name the variable associated with δ_1 after x_1. Suppose that in Ω, x_1 is connected to another variable x_i of some g labeling the vertex v using $\neq_2$.

- If $u = v$, i.e., f and g are labeled on the same vertex in Ω, then by definition the resulting signature $f^{x_1 \neq x_i} = f_{1i}^{10} = h_i^0$ obtained from f by connecting variables x_1 and x_i is in $\mathscr{D}_1 \cup \mathscr{A}$. Then, the instance is reduced to a smaller instance with two fewer edges where f is replaced by $f^{x_1 \neq x_i}$ of smaller arity. Since $f^{x_1 \neq x_i} \in \mathscr{D}_1 \cup \mathscr{A}$, the reduced instance is still an instance of $\#\mathrm{EO}(\mathscr{A} \cup \mathscr{D}_1)$.
- Otherwise, f and g are labeled on two different vertices. By Lemma 12, connecting δ_1 of f with a variable x_i of g, we get a zero signature or a signature $h \otimes \delta_1 \otimes g'$ where $g' = g_{ij}^{01} \in \mathscr{D}_1 \cup \mathscr{A}$ for some $j \in A(g) \backslash \{i\}$. If a zero signature is realized, then clearly we are done. Otherwise, by renaming variables and decomposing $h \otimes \delta_1 \otimes g'$ into two parts $f = h \otimes \delta_1$ and g', again we can reduce the original instance to a smaller instance with two fewer edges where the edges incident to f is changed and g is replaced by g' of smaller arity. Still, since $g' \in \mathscr{D}_1 \cup \mathscr{A}$, the reduced instance is still an instance of $\#\mathrm{EO}(\mathscr{A} \cup \mathscr{D}_1)$.

Thus, in both cases, the instance can be solved recursively and the number of recursions is bounded by $|E|/2$ where E is the edge set of the underlying graph of Ω. ◀

4 Characterization of δ_1-affine δ_0-affine kernels

In this section, we prove the main characterization theorem for δ_1-affine and δ_0-affine kernels using the Hadamard code. We first give an equivalent definition for the δ_1-affine kernel. Note that in Definition 7 we require $h \notin \mathscr{A}$, while in the following lemma we require $\delta_0 \nmid h$.

▶ **Lemma 14.** *An* EO *signature* $f \in K(\mathscr{D}_1)$ *if and only if it satisfies the following two properties:*

1. $f = \delta_1^{\otimes m} \otimes h$ *for some* $m \in \mathbb{Z}_+$ *where* h *is a signature of positive arity,* $\delta_1, \delta_0 \nmid h$*;*
2. *for every* $i \in A(h)$, $h_i^0 \in \mathscr{A}$*.*

Note that the signature f_2 in Example 2 is a δ_1-affine kernel. Trivially, any EO signature of support size 3 with at least one δ_1 factor and no δ_0 factor is a δ_1-affine kernel, since any signature obtained from it by pinning a variable to 0 has a support of size 1 or 2, which is affine. We say $f \in K(\mathscr{D}_1)$ (or $K(\mathscr{D}_0)$) is a trivial kernel if $|\mathscr{S}(f)| = 3$. Below, we prove the characterization theorem for non-trivial δ_1-affine and δ_0-affine kernels, which is essentially equivalent to Theorem 8.

▶ **Theorem 15.** *An* EO *signature* f *is a non-trivial* δ_1*-affine (or* δ_0*-affine) kernel if and only if* $\mathscr{S}(f)$ *is an* m*-multiple of a balanced 1-Hadamard (or balanced 0-Hadamard respectively) code of size* $2^k - 1$ *for some* $k \geq 3$ *and* $m \geq 1$*.*

We give a proof outline for this theorem. First, we prove δ_1-affine kernels admit certain arities and support sizes by carefully counting the number of 0s and 1s in each column of their supports. More specifically, we prove a non-trivial δ_1-affine kernel f must have support size $2^k - 1$ and arity $m2^k$, for some $m, k \in Z_+, k \geq 3$ (Lemma 17). Then, we define basic δ_1-affine kernels (Definition 18) of support size $2^k - 1$ and arity 2^k. We show every non-trivial δ_1-affine kernel is a multiple of some non-trivial basic kernel (Lemma 19). Finally, we characterize the basic δ_1-affine kernel by the balanced 1-Hadamard code $\mathscr{H}_{2^k}^{1b}$. We define a special affine signature called butterfly (Definition 20), serving as a bridge connecting basic kernels and balanced Hadamard codes. We show $\mathscr{H}_{2^k}^{1}$ (or $\mathscr{H}_{2^k}^{0}$) is just a half of butterfly (Lemma 24). We also define left and right wings of butterfly (Definition 21) by removing the all-0 (or all-1) row from a half of butterfly. Notice $\mathscr{H}_{2^k}^{0b}$ (or $\mathscr{H}_{2^k}^{1b}$) is obtained by removing the all-0 (or all-1) codeword from $\mathscr{H}_{2^k}^{1}$. Thus, wings of butterfly coincide with $\mathscr{H}_{2^k}^{0b}$ and $\mathscr{H}_{2^k}^{1b}$ (Lemma 24). Finally, we prove basic kernels are precisely wings of butterfly (Lemma 26 and Lemma 28). Therefore, basic kernels, balanced Hadamard codes and wings of butterfly are precisely the same things.

▶ **Lemma 16.** *Suppose $f \in K(\mathscr{D}_1)$. If $\delta_0 \nmid f^{x_i=1}$ for some $i \in A(f)$, then $f^{x_i=1} \in K(\mathscr{D}_1)$.*

▶ **Lemma 17.** *Suppose that $f = \delta_1^{\otimes m} \otimes h \in K(\mathscr{D}_1)$ where $m \geq 1, \delta_1, \delta_0 \nmid h$, and $|\mathscr{S}(f)| > 3$. Then there exists some $k \geq 2$ such that $|\mathscr{S}(f)| = 2^{k+1} - 1$ and $a(f) = m2^{k+1}$. Moreover, we have $\delta_1 \nmid h_i^0, \delta_0 \nmid h_i^1$ and $\#_i^0(h) = 2^k$ for every $i \in A(h)$.*

▶ **Definition 18** (Basic δ_1-affine kernel)**.** *A basic δ_1-affine kernel f_k of order k (or a k-th basic δ_1-affine kernel) is a δ_1-affine kernel of arity 2^k and support size $2^k - 1$.*

For example, the basic δ_1-affine kernel of order 1 is $f_1 = \delta_1 \otimes \delta_0$ and the basic δ_1-affine kernel of order 2 is $f_2 = \delta_1 \otimes \{(001), (010), (100)\}$.

▶ **Lemma 19.** *An* EO signature of arity $m2^k$ is a non-trivial δ_1-affine kernel if and only if it is an m-multiple of a non-trivial basic δ_1-affine kernel of arity 2^k.

Proof. First we assume $f = \delta_1 \otimes g$ is a non-trivial basic δ_1-affine kernel. Then $|\mathscr{S}(f_{\times m})| = |\mathscr{S}(f)| > 3$. We want to show $f_{\times m} \in K(\mathscr{D}_1)$. Note that $f_{\times m} = \delta_1^{\otimes m} \otimes g_{\times m}$ and $\delta_1, \delta_0 \nmid g_{\times m}$. Pinning 0 to the i-th variable of one copy of g in $g_{\times m}$, each of the remaining $m-1$ copies contributes a δ_0 factor. i.e, $(g_{\times m})_i^0 = \delta_0^{\otimes(m-1)} \otimes (g_i^0)_{\times m}$. Clearly $(g_{\times m})_i^0 \in \mathscr{A}$ since $g_i^0 \in \mathscr{A}$. It follows that $f_{\times m} \in K(\mathscr{D}_1)$.

Conversely, assume $f = \delta_1^{\otimes m} \otimes h$ is a non-trivial δ_1-affine kernel, where $m \in \mathbb{Z}_+, \delta_1, \delta_0 \nmid h$. We want to show there exists a non-trivial basic δ_1-affine kernel f_0 such that $f = (f_0)_{\times m}$. By Lemma 17, we have $|\mathscr{S}(f)| = 2^k - 1$ and $\#_i^0(h) = 2^{k-1}$ for every $i \in A(h)$. Fix any $i \in A(h)$, consider $f^{x_i=0} = \delta_1^{\otimes m} \otimes h^{x_i=0}$. It is an affine EO signature of support size 2^{k-1}. By Lemma 17, $\delta_1 \nmid h^{x_i=0}$. Factoring out δ_0 in $h^{x_i=0}$, we suppose $h^{x_i=0} = \delta_0^{\otimes m_0} \otimes h'$, $\delta_1, \delta_0 \nmid h'$. At the same time, notice h' is constant-weighted. Thus, h' is an EO signature by Lemma 9. Since $f^{x_i=0} = \delta_1^{\otimes m} \otimes \delta_0^{\otimes m_0} \otimes h'$ is also EO, we have $m = m_0$.

On one hand, if x_j is a variable taking constant 0 in $\mathscr{S}(f^{x_i=0})$, then the values of x_j and x_i must be identical in $\mathscr{S}(h)$, because the j-th column already contributes 2^{k-1} many 0s in $\mathscr{S}(f^{x_i=0})$ and the remaining $2^{k-1} - 1$ rows must be all 1s. On the other, if x_j is a variable of h', then x_j and x_i are not identical in $\mathscr{S}(h)$ since they are not identical in $\mathscr{S}(h^{x_i=0})$. Therefore, there are exactly $m_0 = m$ copies of the i-th column in $\mathscr{S}(h)$. Since $i \in A(h)$ is arbitrary, we have $h = (h_0)_{\times m}$ for some signature h_0. Clearly $\delta_1, \delta_0 \nmid h_0$ since $\delta_1, \delta_0 \nmid h$. Notice for every $i \in A(h_0)$, $(h_0^{x_i=0})_{\times m} = ((h_0)_{\times m})^{x_i=0} = h^{x_i=0} = \delta_0 \otimes h_i^0 \in \mathscr{A}$, so $h_0^{x_i=0} \in \mathscr{A}$ and then $(h_0)_i^0 \in \mathscr{A}$. Therefore, $f_0 = \delta_1 \otimes h_0$ is a basic δ_1-affine kernel and $f = (f_0)_{\times m}$. ◀

Below, we show balanced Hadamard codes characterize basic kernels via the bridge of butterfly.

▶ **Definition 20** (Butterfly)**.** *Let $k \geq 1$. An affine signature g of arity 2^{k+1} is a butterfly and denoted by B_k, if it has exactly k free (linearly independent) variables $x_1, x_2, ..., x_k$ and any two different variables do not always take identical values in $\mathscr{S}(g)$.*

The butterfly B_k is unique up to a permutation of variables. It is precisely the affine signature of k free variables with the maximum arity, assuming any two different variables are not identical.

▶ **Definition 21** (Wings of butterfly)**.** *We represent the butterfly B_k by a matrix in the following way. Let $(x_2, x_3, ..., x_{k+1})$ be free variables of B_k, and x_1 be constant 0. Moreover, let $x_m = \lambda_2 x_2 + \lambda_3 x_3 + ... + \lambda_{k+1} x_{k+1}$ and $x_{m+2^k} = x_m + 1$ for $1 \leq m \leq 2^k$, where $\lambda_2, ..., \lambda_{k+1} \in \mathbb{Z}_2$. Through a permutation of rows, we let the first row be $\vec{0}^{2^k} \vec{1}^{2^k}$.*

Then, B_k can be split into the left half and the right half, each of which consisting of 2^k columns. The left wing of B_k, denoted by L_k is the signature obtained form the left half of B_k by removing the first row $\widetilde{0}^{2^k}$. Symmetrically, we define the right wing of B_k denoted by R_k.

▶ Remark 22. Recall that $\widetilde{L_k}$ and $\widehat{R_k}$ are obtained from L_k and R_k by adding an all-0 or all-1 row respectively. They are the left and right halves of B_k respectively which are affine. In fact, $\widetilde{L_k}$ is precisely the affine signature of k free variables with maximum arity, assuming every variable is a linear combination of free variables and any two variables are not identical. While $\widehat{R_k}$ is the complement of $\widetilde{L_k}$.

▶ **Example 23.** We give a butterfly with 2 free variables x_2, x_3.

$$B_2 = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array}.$$

The other variables are decided by the following linear or affine linear combinations. $x_1 = 0x_2+0x_3, x_4 = x_2+x_3, x_5 = 0x_2+0x_3+1, x_6 = x_2+0x_3+1, x_7 = 0x_2+x_3+1, x_8 = x_2+x_3+1$.

The left and right wings of B_2 are $L_2 = \left[\begin{smallmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{smallmatrix}\right]$ and $R_2 = \left[\begin{smallmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{smallmatrix}\right]$.

▶ **Lemma 24.** *As a signature, the balanced 1-Hadamard (or 0-Hadamard) code is the right (or left) wing of butterfly up to a permutation of variables.*

Proof. We only need to prove $\mathscr{H}^0_{2^k}$ is the left half of B_k. Note that $\mathscr{H}^0_{2^k}$ is a linear code, then it is affine as a signature. By the definition of Hadamard matrix and Hadamard code, every two different rows of $\mathscr{H}^0_{2^k}$ do not take identical values. By Sylvester's construction, $\mathscr{H}^0_{2^k}$ is symmetic as a matrix. Thus, every two different variables of $\mathscr{H}^0_{2^k}$ do not take identical values in its support. Because $\mathscr{H}^0_{2^k}$ is of arity 2^k, support size 2^k, and has an all-0 vector in its support, it is precisely the affine signature of k free variables with maximum arity, assuming every variable is a linear combination of free variables and any two variables are not identical. Therefore, $\mathscr{H}^{0b}_{2^k} = L_k$ is just removing the all-0 row of $\mathscr{H}^0_{2^k}$. ◀

So far we show the equivalence of balanced Hadamard code and wings of butterfly. Next we focus on the relationship between basic kernels and wings of butterfly. For convenience, we only consider the basic δ_1-affine kernel and the right wing of butterfly in the following. The other case is symmetric.

▶ **Lemma 25.** *Let $k \geq 3$ and R_k be the right wing of B_k, then R_k is an EO signature. Moreover, every two different rows (or columns) of R_k do not take identical or opposite values.*

Proof. Because $\mathscr{H}^{1b}_{2^k}$ is an EO signature and every two different rows and columns do not take identical or opposite values, this lemma is true because $R_k = \mathscr{H}^{1b}_{2^k}$ by Lemma 24. ◀

▶ **Lemma 26.** *Let R_k be the right wing of a butterfly. Then R_k is the k-th basic δ_1-affine kernel.*

Proof. For $k = 1, 2$ it can be checked by definition. Now we assume $k \geq 3$. First by Lemma 25 R_k is EO. Then we show $R_k \in K(\mathscr{D}_1)$. Notice $R_k = \delta_1 \otimes r_k, \delta_0, \delta_1 \nmid r_k$, and $(r_k)^0_i = (\widehat{r_k})^0_i \in \mathscr{A}$, since $\widehat{r_k} \in \mathscr{A}$. Finally, because R_k is of arity 2^k and support size $2^k - 1$, it is a basic δ_1-affine kernel of order k. ◀

Below, we prove the more complicated direction that a basic δ_1-affine kernel of order k is just R_k up to a permutation of variables. We prove this by induction. We first prove certain properties for the $k+1$-th δ_1-affine kernel assuming the k-th δ_1-affine kernel is unique up to a permutation of variables.

▶ **Lemma 27.** *Suppose $k \in \mathbb{Z}_+, k \geq 2$. If the k-th basic δ_1-affine kernel f_k is unique up to a permutation of variables, then any $(k+1)$-th basic δ_1-affine kernel $f = \delta_1 \otimes h$ satisfies the following properties for every $l \in A(h)$.*

1. *$f^{x_l=1} = (f_k)_{\times 2}$ up to a permutation of variables.*
2. *$f^{x_l=0} = B_k$ up to a permutation of variables.*
3. *Two variables x_i and x_j take opposite values in $\mathscr{S}(f^{x_l=0})$ if and only if they take identical values in $\mathscr{S}(f^{x_l=1})$.*

▶ **Lemma 28.** *The k-th basic δ_1-affine kernel f_k is precisely R_k up to a permutation of variables.*

Proof. For $k = 1$, we have $f_1 = \delta_1 \otimes \delta_0$ and this lemma is clear. For $k = 2$, $f_2 = \delta_1 \otimes \{(001), (010), (100)\}$ and this lemma is true by Example 23. Now assume this lemma holds for positive integers less or equal than $k (k \geq 2)$. By Lemma 27, we know any $(k+1)$-th δ_1-affine kernel $f_{k+1} = \delta_1 \otimes h_{k+1}$ satisfies the following three properties:

$f_{k+1} =$

$0 \cdots 0$	$1 \cdots 1$
$\overline{f_k}$	f_k
f'_k	f'_k

1. Extracting any variable of h_{k+1} to 0, up to a permutation of rows, the first 2^k rows of f_{k+1} is butterfly B_k.
2. Extracting any variable of h_{k+1} to 1, up to a permutation of rows, the last $2^k - 1$ rows is $(f_k)_{\times 2}$. However, when we write the above butterfly into the specific form, we fix the order of columns. Then we denote the last $2^k - 1$ rows by two f'_k, where $f'_k = f_k$ up to a permutation of rows and columns.
3. Every two different variables take identical values in $\mathscr{S}((f_{k+1})^{x_i=1})$ if and only if they take opposite values in $\mathscr{S}((f_{k+1})^{x_i=0})$, $i \in A(h_{k+1})$.

Because every two different columns of f_k (as well as f'_k) do not take identical or opposite values by Lemma 25, we can easily check this proposition still holds for f_{k+1}. Next we show the signature $\widehat{f_{k+1}}$ is affine. We first notice $\widehat{f_{k+1}}$ has the following properties, which holds for every $i \in A(\widehat{f_{k+1}})$:

1. $(\widehat{f_{k+1}})^{x_i=0} = (f_{k+1})^{x_i=0} = B_k \in \mathscr{A}$.
2. $(\widehat{f_{k+1}})^{x_i=1} = (\widehat{f_k})_{\times 2} \in \mathscr{A}$, since f_k is the right wing of B_k by induction hypothesis, hence $\widehat{f_k}$ is the right half of B_k, which is affine.

We prove $\widehat{f_{k+1}} \in \mathscr{A}$ by contradiction. If $\widehat{f_{k+1}} \notin \mathscr{A}$, there exists $\alpha', \beta', \gamma' \in \mathscr{S}(\widehat{f_{k+1}})$, such that $\theta' = \alpha' \oplus \beta' \oplus \gamma' \notin \mathscr{S}(\widehat{f_{k+1}})$. If there exists a position j such that $\alpha'_j = \beta'_j = \gamma'_j = 0$, assume $j = 1$ without loss of generality. Then $\alpha' = 0\alpha, \beta' = 0\beta, \gamma' = 0\gamma, \theta' = 0\theta$, where $\alpha, \beta, \gamma \in \mathscr{S}(\widehat{(f_{k+1})}^0_j)$ and $\theta = \alpha \oplus \beta \oplus \gamma$. Because $(\widehat{f_{k+1}})^0_j \in \mathscr{A}$, we have $\theta \in \mathscr{S}((\widehat{f_{k+1}})^0_j)$.

Then $\theta' = 0\theta \in \mathscr{S}(\widehat{f_{k+1}})$, contradiction! Similarly, if there exists a position j such that $\alpha'_j = \beta'_j = \gamma'_j = 1$, we also get a contradiction. Then, by enumerating all possible combinations of $\alpha', \beta', \gamma' \in \mathscr{S}(\widehat{f_{k+1}})$, we can show $\widehat{f_{k+1}} \in \mathscr{A}$. Because $\mathscr{S}(\widehat{f_{k+1}})$ contains an all-1 vector; $|\mathscr{S}(\widehat{f_{k+1}})| = a(\widehat{f_{k+1}}) = 2^{k+1}$; and there is no identical pair in $\mathscr{S}(\widehat{f_{k+1}})$, f_{k+1} is precisely the right wing of B_{k+1}. ◀

So far by Lemma 26 and Lemma 28 we prove the equivalence between basic kernels and wings of butterfly. Combining with Lemma 24 we get the proof of Theorem 15.

5 Hardness result

Recall that the binary disequality signature $(\neq)$ is $\{10, 01\}$. The symbol "$\leq_T$" stands for Turing reduction.

▶ **Definition 29** (ARS). *A signature f satisfies ARS if $f(\alpha) = f(\overline{\alpha})$ for every $\alpha \in \mathscr{S}(f)$.*

First, we show that $\delta_1 \otimes \delta_0$ is always realizable from an EO signature not satisfying ARS.

▶ **Lemma 30.** *If an* EO signature f does not satisfy ARS, then $\#\mathrm{EO}(f, \delta_1 \otimes \delta_0) \leq_T \#\mathrm{EO}(f)$.

With the help of $\delta_1 \otimes \delta_0$, we have the following reduction.

▶ **Lemma 31.** *If an* EO signature f does not satisfy ARS, then $\#\mathrm{EO}(\{f, f^{xi=1}, f^{xi=0}\}) \leq_T \#\mathrm{EO}(f)$ for every $i \in A(f)$.

Next we will prove $\#\mathrm{EO}(f, g)$ is #P-hard if $f \in \mathscr{D}_1 \backslash \mathscr{A}$ and $g \in \mathscr{D}_0 \backslash \mathscr{A}$, which implies that $\#\mathrm{EO}(\mathscr{D}_1 \cup \mathscr{D}_0)$ is #P-hard. We first consider a basic case where f and g are basic δ_1-affine and δ_0-affine kernels of order 2 respectively.

▶ **Lemma 32.** *Let $f_2(x_1, x_2, x_3, x_4) = \left[\begin{smallmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{smallmatrix}\right]$ and $g_2(y_1, y_2, y_3, y_4) = \left[\begin{smallmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{smallmatrix}\right]$. Then $\#EO(f_2, g_2)$ is #P-hard.*

Proof. Connecting the variables x_1 and y_1, x_2 and y_2 using $\neq$ respectively, we realize a signature h of arity 4 where

$$h = \begin{array}{c} \begin{matrix} x_3 & x_4 & y_3 & y_4 \end{matrix} \\ \left[\begin{matrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{matrix}\right] \end{array}.$$

It is known that $\#\mathrm{EO}(h)$ is #P-hard by the complexity classification for the six-vertex model [7]. Then, $\#\mathrm{EO}(f_2, g_2)$ is #P-hard. ◀

This #P-hardness result can be generalized to any multiples of f_2 and g_2.

▶ **Lemma 33.** *Let $f = (f_2)_{\times a}, g = (g_2)_{\times b}$, where $a, b \in \mathbb{Z}_+$. Then $\#EO(f, g)$ is #P-hard.*

Proof. Connect f and g using $\neq$ block by block. That is, connect variables x_i and y_i using $\neq$ for $1 \leq i \leq 4$, where x_i, y_i are variables of one block f_2 and g_2 as denoted in Lemma 32.

If $a = b = 1$, it is the case in Lemma 32. If $a = b > 1$, we can realize

$$h' = \begin{array}{c} \begin{matrix} x_3 & x_4 & y_3 & y_4 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

which is also #P-hard by the complexity classification of the six vertex model.

If $a \neq b$, we may assume $a > b$ without loss of generality. Then we can realize $f' = (f_2)_{\times(a-b)}$. Then connect f' and g block by block and continue this process. We finally realize $(f_2)_{\times d}$ and $(g_2)_{\times d}$ where $d = gcd(a, b)$ is computed by the Euclidean algorithm. Then it is reduced to the case $a = b$. ◀

This #P-hardness result can still be generalized to any basic δ_1-affine and δ_0-affine kernels of support size 3.

▶ **Lemma 34.** *Let $f \in \mathscr{D}_1 \backslash \mathscr{A}, g \in \mathscr{D}_0 \backslash \mathscr{A}$, and $|\mathscr{S}(f)| = |\mathscr{S}(g)| = 3$. Then $\#EO(f, g)$ is #P-hard.*

▶ **Theorem 35.** *If $f \in \mathscr{D}_1 \backslash \mathscr{A}, g \in \mathscr{D}_0 \backslash \mathscr{A}$, then $\#EO(f, g)$ is #P-hard.*

Proof. We first show we can realize a signature $f' \in \mathscr{D}_1 \backslash \mathscr{A}$ of support size 3 if $|\mathscr{S}(f)| > 3$. If $f \in K(\mathscr{D}_1)$, suppose $f = (f_k)_{\times m}$ by Lemma 19, where f_k is the δ_1-affine kernel of order k. we can extract 1 in some variable x_i of f by Lemma 31 and realize $f^{x_i=1}$, which is a multiple of f_{k-1} by Lemma 27. Continuing this process we can finally realize multiple of f_2. If $f \notin K(\mathscr{D}_1)$, suppose $f = \delta_1 \otimes h$. Then there exists $i \in A(h)$ such that $h_i^0 \in \mathscr{D}_1 \backslash \mathscr{A}$. We have $3 < |\mathscr{S}(h_i^0)| < |\mathscr{S}(f)|$. Repeating the process, either $h_i^0 \in K(\mathscr{D}_1)$ or it can realize a signature in $\mathscr{D}_1 \backslash \mathscr{A}$ of smaller support size. Finally we can realize $f' \in \mathscr{D}_1 \backslash \mathscr{A}$ of support size 3. Similarly, we can realize $g' \in \mathscr{D}_0 \backslash \mathscr{A}$ of support size 3. Then by Lemma 34 we prove $\#\mathrm{EO}(f, g)$ is #P-hard. ◀

References

1. Miriam Backens. A new Holant dichotomy inspired by quantum computation. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.16.
2. Miriam Backens. A complete dichotomy for complex-valued Holant^c. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.12.
3. Norman Biggs, E Keith Lloyd, and Robin J Wilson. *Graph Theory, 1736-1936*. Oxford University Press, 1986.
4. Andrei Bulatov, Martin Dyer, Leslie Ann Goldberg, Markus Jalsenius, and David Richerby. The complexity of weighted Boolean #CSP with mixed signs. *Theoretical Computer Science*, 410(38-40):3949–3961, 2009. doi:10.1016/J.TCS.2009.06.003.
5. Jin-Yi Cai, Zhiguo Fu, and Shuai Shao. Beyond #CSP: A dichotomy for counting weighted eulerian orientations with ars. *Information and Computation*, 275:104589, 2020. doi:10.1016/J.IC.2020.104589.
6. Jin-Yi Cai, Zhiguo Fu, and Shuai Shao. From Holant to quantum entanglement and back. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.22.

7 Jin-Yi Cai, Zhiguo Fu, and Mingji Xia. Complexity classification of the six-vertex model. *Information and Computation*, 259:130–141, 2018. doi:10.1016/J.IC.2018.01.003.

8 Jin-Yi Cai, Heng Guo, and Tyson Williams. A complete dichotomy rises from the capture of vanishing signatures. *SIAM Journal on Computing*, 45(5):1671–1728, 2016. doi:10.1137/15M1049798.

9 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Dichotomy for Holant* problems of Boolean domain. In *Proceedings of the 22nd annual ACM-SIAM symposium on Discrete Algorithms*, pages 1714–1728. SIAM, 2011.

10 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. The complexity of complex weighted Boolean #CSP. *Journal of Computer and System Sciences*, 80(1):217–236, 2014. doi:10.1016/J.JCSS.2013.07.003.

11 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Dichotomy for real Holant^c problems. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1802–1821. SIAM, 2018. doi:10.1137/1.9781611975031.118.

12 Nadia Creignou and Miki Hermann. Complexity of generalized satisfiability counting problems. *Information and computation*, 125(1):1–12, 1996. doi:10.1006/INCO.1996.0016.

13 Martin Dyer, Leslie Ann Goldberg, and Mark Jerrum. The complexity of weighted Boolean #CSP. *SIAM Journal on Computing*, 38(5):1970–1986, 2009. doi:10.1137/070690201.

14 Leonhard Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, pages 128–140, 1741.

15 Michel Las Vergnas. On the evaluation at (3, 3) of the Tutte polynomial of a graph. *Journal of Combinatorial Theory, Series B*, 45(3):367–372, 1988. doi:10.1016/0095-8956(88)90079-2.

16 Jiabao Lin and Hanpin Wang. The complexity of Boolean Holant problems with nonnegative weights. *SIAM Journal on Computing*, 47(3):798–828, 2018. doi:10.1137/17M113304X.

17 Boning Meng, Juqiu Wang, and Mingji Xia. P-time algorithms for typical #EO problems. *arXiv preprint*, 2024. doi:10.48550/arXiv.2410.11557.

18 Milena Mihail and Peter Winkler. On the number of Eulerian orientations of a graph. *Algorithmica*, 16(4-5):402–414, 1996. doi:10.1007/BF01940872.

19 Linus Pauling. The structure and entropy of ice and of other crystals with some randomness of atomic arrangement. *Journal of the American Chemical Society*, 57(12):2680–2684, 1935.

20 Franz Rys. Über ein zweidimensionales klassisches Konfigurationsmodell. In *Helvetica Physica Acta*, volume 36(5), page 537, 1963.

21 Shuai Shao and Jin-Yi Cai. A dichotomy for real Boolean Holant problems. In *Proceedings of the 61st IEEE Annual Symposium on Foundations of Computer Science (FOCS 2020)*, pages 1091–1102. IEEE, 2020. doi:10.1109/FOCS46700.2020.00105.

22 John C Slater. Theory of the transition in KH_2PO_4. *The Journal of Chemical Physics*, 9(1):16–33, 1941.

When to Give up on a Parallel Implementation

Nathan S. Sheffield
MIT, Cambridge, MA, USA

Alek Westover
MIT, Cambridge, MA, USA

Abstract

In the Serial Parallel Decision Problem (SPDP), introduced by Kuszmaul and Westover [SPAA'24], an algorithm receives a series of tasks online, and must choose for each between a serial implementation and a parallelizable (but less efficient) implementation. Kuszmaul and Westover describe three decision models: (1) **Instantly-committing** schedulers must decide on arrival, irrevocably, which implementation of the task to run. (2) **Eventually-committing** schedulers can delay their decision beyond a task's arrival time, but cannot revoke their decision once made. (3) **Never-committing** schedulers are always free to abandon their progress on the task and start over using a different implementation. Kuszmaul and Westover gave a simple instantly-committing scheduler whose total completion time is 3-competitive with the offline optimal schedule, and proved two lower bounds: no eventually-committing scheduler can have competitive ratio better than $\phi \approx 1.618$ in general, and no instantly-committing scheduler can have competitive ratio better than 2 in general. They conjectured that the three decision models should admit different competitive ratios, but left upper bounds below 3 in any model as an open problem.

In this paper, we show that the powers of instantly, eventually, and never committing schedulers are distinct, at least in the "massively parallel regime". The massively parallel regime of the SPDP is the special case where the number of available processors is asymptotically larger than the number of tasks to process, meaning that the *work* associated with running a task in serial is negligible compared to its *runtime*. In this regime, we show (1) The optimal competitive ratio for instantly-committing schedulers is 2, (2) The optimal competitive ratio for eventually-committing schedulers lies in $[1.618, 1.678]$, (3) The optimal competitive ratio for never-committing schedulers lies in $[1.366, 1.500]$. We additionally show that our instantly-committing scheduler is also 2-competitive outside of the massively parallel regime, giving proof-of-concept that results in the massively parallel regime can be translated to hold with fewer processors.

2012 ACM Subject Classification Theory of computation → Online algorithms

Keywords and phrases Scheduling, Multi-Processor, Online-Algorithms

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.87

Related Version *arxiv*: `https://arxiv.org/abs/2408.16092`

1 Introduction

1.1 Background

Many computational tasks can be performed quickly in parallel over a large number of processors – but such parallel implementations may be less work-efficient than a serial implementation on a single processor, requiring substantially more total computation time across all machines. When several different tasks must be completed in as little total time as possible, this trade-off between work and time can necessitate running different tasks in different modes: small tasks can be done in serial to save work, while large tasks must be parallelized to prevent their serial runtimes from dominating the overall computation.

To formalize this problem, Kuszmaul and Westover introduced the **Serial Parallel Decision Problem (SPDP)** [16]. In their model, each task has exactly two possible implementations: an "embarrassingly parallel" implementation which can be worked on by


© Nathan S. Sheffield and Alek Westover;
licensed under Creative Commons License CC-BY 4.0



multiple machines at once (where the rate of progress on the implementation is proportional to the number of processors assigned to it), and a serial implementation which can only be worked on by a single processor at a time. If all tasks are available at time 0, it is easy to efficiently determine the optimal strategy: all jobs with serial completion time smaller than some threshold can be run in serial, and the larger tasks must be run in parallel. The model becomes interesting when previously-unknown tasks are allowed to arrive at arbitrary times, and one wishes to minimize the competitive ratio between the total completion time of an *online* algorithm compared to the offline optimal completion time.

Kuszmaul and Westover define three distinct versions of this model, parameterized by the degree to which the online scheduler is able to reverse its decisions.

1. An **instantly-committing** scheduler must choose an implementation for each task as soon as the task arrives, and is not allowed to revisit this choice.
2. An **eventually-committing** scheduler may delay choosing an implementation, but must choose one irrevocably before assigning its work to a processor.
3. A **never-committing** scheduler can, at any time, discard all as-yet completed work on an implementation and re-start the task with the other implementation.

The distinction between the eventually- and never-committing models is motivated by potential practical concerns: if a task involves mutating an input in memory, it may not be feasible to cancel an implementation once it begins running. Westover and Kuszmaul present an instantly-committing scheduler achieving competitive ratio 3, and show competitive ratio lower bounds of 2 and $\phi \approx 1.618$ in the instantly-committing and eventually-committing models, respectively (for deterministic schedulers). They conjecture that the ability to delay or cancel choices should allow for more competitive online algorithms, but leave open the problem of finding better competitive ratio upper bounds than 3.

1.2 This Work

In this work, we consider Kuszmaul and Westover's SPDP when the number of available processors is much larger than the number of tasks, noting that all of their upper and lower bounds hold in this parameter regime. This is a particularly simple setting, since the work associated to a serial implementation is now negligible compared to its completion time – running a task in serial means accepting a lower bound on completion time, but requires essentially no work. We can think of this setting as an unrelated-machines scheduling problem with an unlimited number of identical "slow" machines, and a single unrelated "fast" machine, representing a massively parallel implementation of the task across many processors – note that this could also describe scenarios with a literal fast machine, such as a single piece of accelerated hardware. Although we consider the number of processors to be large, we won't assume that tasks complete instantly when run on all the processors; instead, we'll let the total parallel work of each tasks be proportional to p.

Our main results are tight bounds on the competitive ratio of instantly-committing schedulers in this regime, and separations between the strength of all 3 models. Our results are summarized in Table 1. Note that we focus on deterministic schedulers (although we briefly discuss randomized schedulers in the appendix).

In each case, the upper bound comes from a simple heuristic in which the algorithm compares its projected completion time to its current estimate of the optimal completion time. More precisely, at each time t, our scheduler computes an offline optimal strategy "$\mathsf{opt}(t)$" on the truncation of the task sequence to tasks that arrive before time t, and makes decisions based on the completion time of $\mathsf{opt}(t)$.

Table 1 Main Results. * = this work.

Model	Lower Bound	Upper Bound
Instantly-Committing Schedulers	2 [16]	**2** *
Eventually-Committing Schedulers	1.618 [16]	**1.678** *
Never-Committing Schedulers	**1.366** *	**1.5** *

Our main technical contribution is the analysis of the schedulers. Working with $\mathsf{opt}(t)$ is challenging, because the schedules $\mathsf{opt}(t)$ and $\mathsf{opt}(t')$ can be quite different for $t \neq t'$. For instantly-committing schedulers we use an invariant-based approach to bound, at all times, the work taken by our scheduler in terms of the minimum work and completion time among all schedulers. For eventually- and never-committing schedulers this approach is no longer feasible: there is no well defined notion of the "work taken" by our scheduler, because the scheduler may not have committed to a decision yet. Instead, these analyses rely on choosing a couple of critical times to observe the state of our scheduler and $\mathsf{opt}(t)$, and then establishing a dichotomy: either (1) "real fast tasks" (tasks that opt runs on the fast machine) arrive quickly, in which case our scheduler prioritizes real fast tasks on the fast machine and will run most other tasks on slow machines, or (2) real fast tasks arrive slowly, in which case our scheduler never falls too far behind opt, despite making suboptimal use of the fast machine.

In addition to these results, we show that with some effort our instantly-committing scheduler can be adapted to work for any number of processors, fully resolving the question of the optimal competitive ratio of instantly-committing schedulers in the general SPDP, and giving a proof-of-concept that results for a large number of processors can be adapted to hold when the work associated with serial tasks is also a concern.

1.3 Related Work

There is a long line of work studying the phenomenon of work-inefficient parallel implementations in multi-processor scheduling. Typically, the models of limited parallelism considered involve one of three types of jobs:

1. **Rigid** jobs, which come with a number p_i specifying a fixed number of processors the job must be run on at each timestep of its execution.
2. **Moldable** jobs, where the scheduler may choose the (fixed) number of processors the job is run on, and the amount of work scales depending on this choice according to some **speedup curve**.
3. **Malleable** jobs, which like moldable jobs have an associated speedup curve, but where the job may be assigned to different numbers of processors at different timesteps (as opposed to the scheduler choosing a fixed value at the start of the task's runtime).

In each of these cases, there is interest in minimizing the total completion time (**makespan**) in both the offline setting – where problems tend to be NP-hard, but may have approximation algorithms [25, 20, 18, 24, 23] – and the online setting, where the goal is to minimize competitive ratio [8, 7, 4, 14, 9, 29, 28]. Kuszmaul and Westover's Serial Parallel Decision Problem is related to this line of work, but doesn't quite fit into the usual framework – in their model, instead of dealing with an arbitrary speedup curve, there is a single binary decision between a completely serial and perfectly parallelizable implementation.

As noted, the massively-parallel regime of the SPDP considered in this paper can be naturally viewed as a scheduling problem with an unlimited number of identical "slow" machines, and a single unrelated "fast" machine. Standard scheduling problems in the

unrelated machines model have also been well-studied, in terms of both offline approximation algorithms and hardness results [13, 17, 22, 26, 19, 21, 10, 15, 6], and online algorithms [3, 2, 22, 5, 1, 11, 12, 30]. We note, however, that since we treat "slow" machines as an unbounded resource, and there is only a single fast machine available, most of the typical difficulties of multi-processor scheduling problems do not arise. In particular, unlike a typical load-balancing problem where NP-hardness follows from a standard set-partition reduction, the One-Fast-Many-Slow Decision Problem (without dependencies) is easily solvable offline, simply by putting all tasks which finish below a certain threshold on distinct slow machines.

1.4 Open Questions

We leave three main open questions as directions for future work.

▶ **Question 1.** What are the optimal competitive ratios for eventually/never-committing schedulers?

In Section A we identify barriers, showing that improving on our eventually/never-committing schedulers will require substantially different algorithms – but we suspect that such improvements may be possible.

▶ **Question 2.** Are randomized schedulers more powerful than deterministic schedulers?

In the main body of the paper we consider only deterministic schedulers; however, for many online problems randomized algorithms can do substantially better than deterministic ones. In Section C we give some lower bounds against randomized schedulers, but these bounds are weaker than those known for deterministic schedulers.

▶ **Question 3.** Is there a general transformation between schedulers for the massively parallel regime (i.e. the One-Fast-Many-Slow Decision Problem) and the general SPDP?

The fundamental difficulty of the SPDP is deciding between implementations which take a lot of work, and implementations which take a lot of time. This tradeoff is absolute in the massively parallel regime, since the large number of processors means the amount of work associated with a serial implementation is negligible, whereas in the general SPDP it is possible for all processors to be saturated with serial implementations to run. Intuitively, one might expect that having work associated to the serial implementations only makes the problem *easier*, since it makes the tradeoff less dramatic – indeed, Kuszmaul and Westover's competitive ratio lower bounds become weaker when the number of processors is small. So, one might hope that algorithms in the massively parallel regime can be generically translated to limited-processor settings. Formalizing this connection is an interesting direction for future research.

2 Preliminaries

2.1 The One-Fast-Many-Slow Decision Problem

In this section we formally define the **One-Fast-Many-Slow Decision Problem**, where the goal is to distribute work between a single **fast machine** and an unlimited number of **slow machines**. An instance of the problem is a **Task Arrival Process (TAP)** $\mathcal{T} = (\tau_1, \ldots, \tau_n)$, where each task τ_i consists of a tuple (σ_i, π_i, t_i) indicating runtime on a slow machine, runtime on the fast machine, and arrival time, respectively, such that $t_1 \leq \cdots \leq t_n$ and such that

$\sigma_i \leq \pi_i$. A valid schedule associates at most one task to each machine at each point in time[1] such that no work is done on any task before its arrival time, each task runs on at most 1 machine, and each task τ_i is either run for a total of σ_i time on some slow machine, or a total of π_i time on the fast machine. The **completion time** (also known as **makespan**) of the schedule is the time when the last task is finished.

We will be interested in online algorithms for this problem. An online scheduler learns about each task only at its arrival time, and at each time t must already have fixed the prefix of the schedule on times less than t. We define three distinct models for how these online decisions are made:

1. For each task τ, an **instantly-committing** scheduler must fix at τ's arrival time the machine that τ will run on.
2. An **eventually-committing** scheduler need not fix a machine for any task until that task begins running.
3. A **never-committing** scheduler is an eventually-committing scheduler with the additional power to, at any time, "cancel" a task from the schedule, erasing all work previously done on the task and allowing it to be re-assigned to a new machine.

In each case, we are interested in minimizing the **competitive ratio** of an online scheduler, which is the supremum over all TAPs of the ratio of the online scheduler's completion time to the completion time of an optimal scheduler on that TAP.

2.2 Connection to the SPDP

In Kuszmaul and Westover's Serial Parallel Decision Problem, a scheduler must allocate work to p equally-powerful processors, where each task is specified by the work of the serial implementation, the work of the parallel implementation, and the runtime. The scheduler must choose whether to run each task in serial or parallel, and then must assign the resulting work to the p processors, where parallel work can run on multiple processors at once but serial work cannot.

We can define the **massively parallel** regime of this problem to be the limit as the number of processors becomes large compared to the number of tasks. Letting n be the number of tasks, if $n \leq \varepsilon p$ then restricting the serial implementations to run on only the first n many processors, and the parallel implementations to run on only the last $p - n$ many processors, the completion time can increase by at most a $\frac{1}{1-\varepsilon}$ factor. This corresponds directly to the One-Fast-Many-Slow Decision Problem: we think of each of these serial processors as a "slow machine", noting that since we have as many processors as we have tasks there are effectively an unlimited number of processors. We think of the parallel processors collectively as a "fast machine", noting that we can assume without loss of generality that, at any point in time, all parallel processors are running the same parallel implementation.

2.3 Notation

We now introduce our notation for describing and analyzing schedulers. For algorithm alg and TAP $\mathcal{T}$, we let $\mathsf{C}_{\mathsf{alg}}^{\mathcal{T}}$ be the completion time of alg on $\mathcal{T}$. Let $\mathcal{T}^t$ be the truncation of TAP $\mathcal{T}$ consisting of the tasks τ_i with $t_i \leq t$. When $\mathcal{T}$ is clear from context we will write $\mathsf{C}_{\mathsf{alg}}^{t}$ to denote $\mathsf{C}_{\mathsf{alg}}^{\mathcal{T}^t}$, and we will write $\mathsf{C}_{\mathsf{alg}}$ to denote $\mathsf{C}_{\mathsf{alg}}^{\infty} = \mathsf{C}_{\mathsf{alg}}^{\mathcal{T}}$. We will also use $\widetilde{\mathsf{C}}_{\mathsf{alg}}$ to denote the completion time of the fast machine – that is, the final time when the fast machine has work.

[1] In order for the notions like "amount of work performed on τ_i" to be well-defined, we must additionally mandate that a schedule be measurable. Alternatively, one can assume that time is discretized into appropriately fine timesteps.

It will be useful to be able to talk about the optimal completion time of a prefix of the TAP. Define the schedule $\mathsf{opt}(t)$ to be a schedule for $\mathcal{T}^t$ with minimal completion time. Note that $\mathsf{opt}(t)$ is only defined as an offline strategy, but that an online algorithm can compute it (efficiently!) at time t, thus obtaining a lower bound on $\mathsf{C}_{\mathsf{opt}}$, which will be useful to inform the algorithm's future decisions. For ease of notation, we'll often abbreviate $\mathsf{C}^t_{\mathsf{opt}(t)}$ as C^t.

There may be many sets of decisions which result in the optimal completion time; as opposed to letting $\mathsf{opt}(t)$ be an arbitrary such scheduler, it will be useful to fix a canonical one, which we will do by letting $\mathsf{opt}(t)$ run as many tasks in serial as possible.

▶ **Scheduler 4.** The scheduler $\mathsf{opt}(t)$, defined on $\mathcal{T}^t$, makes decisions as follows:

- If τ_i has $\sigma_i + t_i \leq \mathsf{C}^t$, run τ_i on a slow machine when it arrives.
- Otherwise, run τ_i on the fast machine. Prioritize tasks with larger $\sigma_i + t_i$, and break ties by taking tasks with smaller i.

Finally, we let $[n] = \{1, \ldots, n\}$, and for a set J of tasks we will write π_J to denote $\sum_{j \in J} \pi_j$.

3 A 2-Competitive Instantly-Committing Scheduler

In this section we present and analyze a 2-competitive instantly-committing scheduler. Kuszmaul and Westover showed that a competitive ratio of $(2 - \varepsilon)$ is impossible for instantly-committing schedulers, so our scheduler is optimal. The scheduler, which we call ins ("*ins*tantly commiting") is defined in Scheduler 5.

▶ **Scheduler 5.** When task τ_i arrives:

- If $\sigma_i + t_i > 2\mathsf{C}^{t_i}$ run τ_i on the fast machine, with the fast machine processing tasks in order of arrival.
- Otherwise run τ_i on a slow machine.

We analyze ins by showing inductively that $\widetilde{\mathsf{C}}_{\mathsf{ins}}$ (the completion time of the fast machine) is small compared to the work and completion time of *any* other schedule. For length n TAP $\mathcal{T}$, scheduler alg, and $i \in [n]$, we define the quantity $\mathsf{K}^t_{\mathsf{alg}}$ to be the sum of π_j for all tasks $\tau_j \in \mathcal{T}^t$ that alg runs on the fast machine. The key to analyzing ins is the following lemma.

▶ **Lemma 6.** *Fix a length n TAP. For all $i \in [n]$, and for all instantly-committing schedulers* alg*,*

$$\widetilde{\mathsf{C}}^{t_i}_{\mathsf{ins}} \leq \mathsf{C}^{t_i}_{\mathsf{alg}} + \mathsf{K}^{t_i}_{\mathsf{alg}}. \tag{1}$$

Proof. We prove the lemma by induction on i. For $i = 1$ the claim is trivial. Now, fix $i \in [n-1]$, alg and assume the lemma for i and for all alg'; we will prove the lemma for $i+1$, alg.

If ins runs τ_{i+1} on a slow machine then $\widetilde{\mathsf{C}}^{t_{i+1}}_{\mathsf{ins}} = \widetilde{\mathsf{C}}^{t_i}_{\mathsf{ins}}$, and $\mathsf{C}^{t_i}_{\mathsf{alg}} + \mathsf{K}^{i}_{\mathsf{alg}} \leq \mathsf{C}^{t_{i+1}}_{\mathsf{alg}} + \mathsf{K}^{i+1}_{\mathsf{alg}}$. Thus, the invariant Equation (1) is maintained. We always have $\widetilde{\mathsf{C}}^{t_{i+1}}_{\mathsf{ins}} \leq \widetilde{\mathsf{C}}^{t_i}_{\mathsf{ins}} + \pi_{i+1}$, so if alg runs τ_{i+1} on the fast machine then the invariant Equation (1) is also maintained, since then $\mathsf{C}^{t_{i+1}}_{\mathsf{alg}} + \mathsf{K}^{t_{i+1}}_{\mathsf{alg}} \geq \mathsf{C}^{t_i}_{\mathsf{alg}} + \mathsf{K}^{t_i}_{\mathsf{alg}} + \pi_{i+1} \geq \widetilde{\mathsf{C}}^{t_i}_{\mathsf{ins}} + \pi_{i+1}$ by the inductive hypothesis.

The final case to consider is when alg runs τ_{i+1} on a slow machine, while ins runs τ_{i+1} on the fast machine. From the definition Scheduler 5 of ins, the fact that ins ran τ_{i+1} on the fast machine implies

$$2\mathsf{C}^{t_{i+1}} < \sigma_{i+1} + t_{i+1}. \tag{2}$$

On the other hand, alg ran τ_{i+1} on the fast machine. Thus,

$$\sigma_{i+1} + t_{i+1} \leq \mathsf{C}_{\mathsf{alg}}^{t_{i+1}}. \tag{3}$$

Now, we use the invariant for $(i, \mathsf{opt}_{t_{i+1}})$ to bound $\widetilde{\mathsf{C}}_{\mathsf{ins}}^{t_{i+1}}$. We have:

$$\widetilde{\mathsf{C}}_{\mathsf{ins}}^{t_{i+1}} \leq \widetilde{\mathsf{C}}_{\mathsf{ins}}^{t_i} + \pi_{i+1} \leq \mathsf{K}_{\mathsf{opt}(t_{i+1})}^{t_i} + \mathsf{C}_{\mathsf{opt}(t_{i+1})}^{t_i} + \pi_{i+1}. \tag{4}$$

Because of Equation (2) we know that $\mathsf{opt}(t_{i+1})$ must run τ_{i+1} on the fast machine. So, we have

$$\mathsf{K}_{\mathsf{opt}(t_{i+1})}^{t_i} + \mathsf{C}_{\mathsf{opt}(t_{i+1})}^{t_i} + \pi_{i+1} = \mathsf{K}_{\mathsf{opt}(t_{i+1})}^{t_{i+1}} + \mathsf{C}_{\mathsf{opt}(t_{i+1})}^{t_i} \leq 2\mathsf{C}^{t_{i+1}}. \tag{5}$$

Stringing together the above inequalities Equation (4), Equation (5), Equation (2), and Equation (3), we get

$$\widetilde{\mathsf{C}}_{\mathsf{ins}}^{t_{i+1}} < \mathsf{C}_{\mathsf{alg}}^{t_{i+1}}.$$

Thus, the invariant Equation (1) holds. ◀

Using Lemma 6 it is easy to show that ins is 2-competitive.

▶ **Theorem 7.** ins *is a 2-competitive instantly-committing scheduler.*

Proof. By Lemma 6 we have $\widetilde{\mathsf{C}}_{\mathsf{ins}}^{t_n} \leq 2\mathsf{C}_{\mathsf{opt}}$. Thus, ins finishes using the fast machine before time $2\mathsf{C}_{\mathsf{opt}}$. Any task that ins runs on a slow machine must have $\sigma_i + t_i \leq 2\mathsf{C}_{\mathsf{opt}}$, so these tasks finish before $2\mathsf{C}_{\mathsf{opt}}$ as well. ◀

4 A 1.678-Competitive Eventually-Committing Scheduler

In this section we present and analyze a ξ-competitive eventually-committing scheduler, where $\xi \approx 1.678$ is the real root of the polynomial $2x^3 - 3x^2 - 1$. Kuszmaul and Westover gave a lower bound of $\phi \approx 1.618$ on the competitive ratio of any eventually-committing scheduler and conjectured that this lower bound is tight. Our scheduler represents substantial progress towards resolving Kuszmaul and Westover's conjecture, improving on their previous best algorithm which had a competitive ratio of 3. Our scheduler, which we call eve ("*eve*ntually committing"), is defined in Scheduler 8.

▶ **Scheduler 8.** At time t:

- If task τ_i, which has arrived but not yet been started, has $\sigma_i + t \leq \xi\mathsf{C}^t$, then start τ_i on a slow machine.
- Maintain up to one **active** task at a time. The fast machine is always allocated to the active task.
- When there is no active task, but there are unstarted tasks present, choose as the new active task the unstarted task with the largest $\sigma_i + t_i$ value (breaking ties arbitrarily).

▶ **Theorem 9.** eve *is a ξ-competitive eventually-committing scheduler.*

Proof. Fix TAP $\mathcal{T}$. Let $\widetilde{\mathsf{C}}_{\mathsf{eve}}$ denote the time when eve completes the last task run on the fast machine. If τ_i is run on a slow machine at any time t, then τ_i finishes before $\xi\mathsf{C}^t \leq \xi\mathsf{C}_{\mathsf{opt}}$. Thus, it suffices to show that $\widetilde{\mathsf{C}}_{\mathsf{eve}} \leq \xi\mathsf{C}_{\mathsf{opt}}$.

For any $x \in [0, \mathsf{C}_{\mathsf{opt}}]$, let $R(x)$ be the first time that an online algorithm becomes aware that the optimal schedule requires at least x completion time – that is, $R(x) = \inf\{t : \mathsf{C}^t \geq x\}$. Let $\mathcal{A}$ ("actual") be the set of tasks that opt runs on the fast machine, and $\mathcal{F}$ ("fake") be the set of tasks that opt runs on a slow machine but eve runs on the fast machine. We can bound the sizes and arrival times of tasks in $\mathcal{F}$ as follows.

▷ Claim 10. All tasks $\tau_i \in \mathcal{F}$ arrive before time $R(\mathsf{C}_{\mathsf{opt}}/\xi)$, and have $\pi_i < \sigma_i/\xi$.

Proof. All tasks $\tau_i \in \mathcal{F}$ are run on the fast machine by eve, and on slow machines by opt. In particular this means

$$\xi \mathsf{C}^{t_i} < \sigma_i + t_i \leq \mathsf{C}_{\mathsf{opt}} \leq \xi \mathsf{C}^{R(\mathsf{C}_{\mathsf{opt}}/\xi)}.$$

Thus, $t_i < R(\mathsf{C}_{\mathsf{opt}}/\xi)$. To show $\pi_i < \sigma_i/\xi$, note that $\pi_i + t_i \leq \mathsf{C}^{t_i} < \frac{\sigma_i + t_i}{\xi}$. ◁

To analyze when tasks in $\mathcal{F}$ get run it will be useful to partition $\mathcal{F}$ into $\mathcal{F}_{\mathsf{big}} = \{\tau_i \in \mathcal{F} \colon \sigma_i + t_i > \mathsf{C}_{\mathsf{opt}}/\xi\}$ and $\mathcal{F}_{\mathsf{small}} = \{\tau_i \in \mathcal{F} \colon \sigma_i + t_i \leq \mathsf{C}_{\mathsf{opt}}/\xi\}$. Now we show that, without loss of generality, eve does not start any tasks in $\mathcal{F}_{\mathsf{small}}$ too late.

▷ Claim 11. If eve starts a task $\tau \in \mathcal{F}_{\mathsf{small}}$ at any time $t \geq R(\mathsf{C}_{\mathsf{opt}}/\xi)$, then $\widetilde{\mathsf{C}}_{\mathsf{eve}} \leq \xi \mathsf{C}_{\mathsf{opt}}$.

Proof. Note that no task $\tau_i \in \mathcal{F}_{\mathsf{small}}$ can be started after time $\mathsf{C}_{\mathsf{opt}}$: since $\mathsf{C}_{\mathsf{opt}} + \sigma_i \leq \mathsf{C}_{\mathsf{opt}} + \mathsf{C}_{\mathsf{opt}}/\xi < \xi \mathsf{C}_{\mathsf{opt}}$, any task $\tau_i \in \mathcal{F}_{\mathsf{small}}$ present but not already running at time $\mathsf{C}_{\mathsf{opt}}$ would be run on a slow machine. Let t_* be the last time after $R(\mathsf{C}_{\mathsf{opt}}/\xi)$ when eve starts a task $\tau \in \mathcal{F}$. If t_* does not exist the claim is vacuously true. In light of our previous observation, $t_* < \mathsf{C}_{\mathsf{opt}}$. Let τ_i be the task that eve starts at time t_*. Because eve prioritizes making tasks with larger $\sigma_j + t_j$ values active, at time t_* there are no tasks $\tau \in \mathcal{F}_{\mathsf{big}} \cup \mathcal{A}$ present. After time t_*, no more tasks from $\mathcal{F}$ can arrive by Claim 10, and at most $\mathsf{C}_{\mathsf{opt}} - t_*$ work in $\mathcal{A}$ can arrive because opt must be able to complete this work. Thus,

$$\widetilde{\mathsf{C}}_{\mathsf{eve}} \leq t_* + (\mathsf{C}_{\mathsf{opt}} - t_*) + \pi_i = \mathsf{C}_{\mathsf{opt}} + \pi_i. \tag{6}$$

Now, because $\tau_i \in \mathcal{F}_{\mathsf{small}}$ we have $\pi_i \leq \mathsf{C}_{\mathsf{opt}}/\xi^2$; using this in Equation (6) we find $\widetilde{\mathsf{C}}_{\mathsf{eve}} \leq (1 + 1/\xi^2)\mathsf{C}_{\mathsf{opt}} \leq \xi \mathsf{C}_{\mathsf{opt}}$. ◁

This means that we can assume that, after time $R(\mathsf{C}_{\mathsf{opt}}/\xi)$, the only tasks that eve runs on the fast machine are $\mathcal{A}$, $\mathcal{F}_{\mathsf{big}}$, and whatever the active task was at time $R(\mathsf{C}_{\mathsf{opt}}/\xi)$. We call the active task at time $R(\mathsf{C}_{\mathsf{opt}}/\xi)$, if one exists, the **stuck** task, denoted τ_s. We split into cases depending on how large this stuck task is.

Case 1. There is no stuck task.

In this case, we in fact have $\widetilde{C}_{\mathsf{eve}} \leq \mathsf{C}_{\mathsf{opt}}$. Since there is no active task at time $R(\mathsf{C}_{\mathsf{opt}}/\xi)$, there are no tasks present but not started on slow machines. By Claim 10, eve will run all tasks $\tau \notin \mathcal{A}$ arriving after time $R(\mathsf{C}_{\mathsf{opt}}/\xi)$ on slow machines. Thus, at all time steps $t \in [R(\mathsf{C}_{\mathsf{opt}}/\xi), \mathsf{C}_{\mathsf{opt}}]$, eve either has no active task on the fast machine, or has some $\tau \in \mathcal{A}$ as the active task on the fast machine, so eve completes $\mathcal{A}$ at least as quickly as opt.

Case 2. There is a stuck task, with $\sigma_s + t_s > \mathsf{C}_{\mathsf{opt}}/\xi$.

Define $\mathcal{A}_{\text{early}} = \{\tau_i \in \mathcal{A} : t_i < R(\mathsf{C}_{\mathsf{opt}}/\xi)\}$ and $\mathcal{A}_{\text{late}} = \mathcal{A} \setminus \mathcal{A}_{\text{early}}$. Let $t < R(\mathsf{C}_{\mathsf{opt}}/\xi)$ be a time when all tasks in $\{\tau_s\} \cup \mathcal{F}_{\mathsf{big}} \cup \mathcal{A}_{\text{early}}$ have already arrived; such a time must exist by Claim 10. Observe that $\mathsf{opt}(t)$ runs all tasks in $\{\tau_s\} \cup \mathcal{F}_{\mathsf{big}} \cup \mathcal{A}_{\text{early}}$ on the fast machine due to $\mathsf{C}^t < \mathsf{C}_{\mathsf{opt}}/\xi$. This further implies that $\pi_{\mathcal{F}_{\mathsf{big}} \cup \{\tau_s\} \cup \mathcal{A}_{\text{early}}} \leq \mathsf{C}_{\mathsf{opt}}/\xi$. Also $\pi_{\mathcal{A}_{\text{late}}} \leq \mathsf{C}_{\mathsf{opt}} - R(\mathsf{C}_{\mathsf{opt}}/\xi)$, simply because opt must complete the work on tasks $\mathcal{A}_{\text{late}}$ after these tasks arrive. By Claim 11 we may assume without loss of generality that, after time

$R(\mathsf{C}_{\mathsf{opt}}/\xi)$, eve is always running a task from $\{\tau_s\} \cup \mathcal{A} \cup \mathcal{F}_{\mathsf{big}}$. Thus,

$$\begin{aligned}\widetilde{\mathsf{C}}_{\mathsf{eve}} &\leq R(\mathsf{C}_{\mathsf{opt}}/\xi) + \pi_{\mathcal{A}\cup\mathcal{F}_{\mathsf{big}}\cup\{\tau_s\}} \\ &= R(\mathsf{C}_{\mathsf{opt}}/\xi) + \pi_{\mathcal{A}_{\mathrm{early}}\cup\mathcal{F}_{\mathsf{big}}\cup\{\tau_s\}} + \pi_{\mathcal{A}_{\mathrm{late}}} \\ &\leq R(\mathsf{C}_{\mathsf{opt}}/\xi) + \mathsf{C}_{\mathsf{opt}}/\xi + (\mathsf{C}_{\mathsf{opt}} - R(\mathsf{C}_{\mathsf{opt}}/\xi)) \\ &\leq \xi\mathsf{C}_{\mathsf{opt}}\end{aligned}$$

Case 3. There is a stuck task, with $\sigma_s + t_s \leq C_{\mathsf{opt}}/\xi$.

This case will be the most difficult to handle of the three. It will be useful to focus now on the tasks of $\mathcal{F}_{\mathsf{big}}$ that arrive after the stuck task is started. Let t_* be the time when eve starts running τ_s, and let $\mathcal{F}_{\mathsf{big}}{}' = \{\tau_i \in \mathcal{F}_{\mathsf{big}} : t_i \geq t_*\}$ be the fake tasks arriving after t_*. We first observe that, if no such tasks arrive, eve performs very well.

▷ Claim 12. If $\mathcal{F}_{\mathsf{big}}{}' = \varnothing$ then $\widetilde{\mathsf{C}}_{\mathsf{eve}} \leq \xi\mathsf{C}_{\mathsf{opt}}$.

Proof. At time $t_\star$ no tasks $\tau_i \in \mathcal{A}\cup\mathcal{F}_{\mathsf{big}}$ can be present, since all such tasks have $\sigma_i + t_i > \mathsf{C}_{\mathsf{opt}}/\xi$, so eve would prioritize running them on the fast machine instead of the stuck task. By Claim 11, we know that after time t_* eve will always be running tasks from $\{\tau_s\} \cup \mathcal{A} \cup \mathcal{F}_{\mathsf{big}}{}'$. The total work on tasks from $\mathcal{A}$ that arrives after time $t_\star$ is at most $\mathsf{C}_{\mathsf{opt}} - t_\star$, so if $\mathcal{F}_{\mathsf{big}}{}' = \varnothing$ we have

$$\widetilde{\mathsf{C}}_{\mathsf{eve}} \leq t_\star + \pi_s + \mathsf{C}_{\mathsf{opt}} - t_\star \leq \sigma_s/\xi + \mathsf{C}_{\mathsf{opt}} \leq \xi\mathsf{C}_{\mathsf{opt}}. \qquad \triangleleft$$

By Claim 12 we may assume $\mathcal{F}_{\mathsf{big}}{}' \neq \varnothing$. So, let $\sigma_{\min} = \min_{\tau_i \in \mathcal{F}_{\mathsf{big}}{}'}(\sigma_i)$; we will be able to control how much work arrives in the TAP by the fact that eve never decides to run the task $\tau_i \in \mathcal{F}_{\mathsf{big}}{}'$ with $\sigma_i = \sigma_{\min}$ on a slow machine. Split $\mathcal{A}$ into $\mathcal{A}_{\mathrm{early}} = \{\tau \in \mathcal{A} : t_\star \leq t_i < R(\sigma_{\min})\}$ and $\mathcal{A}_{\mathrm{late}} = \{\tau \in \mathcal{A} : t_i \geq \max(t_\star, R(\sigma_{\min}))\}$ (note that we use a different threshold to define earliness here than we did in case 2). First we need the following analogue of Claim 11.

▷ Claim 13. If eve starts a task $\tau \in \mathcal{F}_{\mathsf{big}}{}'$ at any time $t \in [R(\mathsf{C}_{\mathsf{opt}}/\xi), \mathsf{C}_{\mathsf{opt}}]$, then $\widetilde{\mathsf{C}} \leq \xi\mathsf{C}_{\mathsf{opt}}$.

Proof. Let t_* be the last time in $[R(\mathsf{C}_{\mathsf{opt}}/\xi), \mathsf{C}_{\mathsf{opt}}]$ when eve starts a task $\tau \in \mathcal{F}'$. If t_* does not exist the claim is vacuously true. Let τ_i be the task that eve starts at time t_*. Because eve prioritizes making tasks with larger $\sigma_j + t_j$ values active, at time t_* there are no tasks $\tau \in \mathcal{A}$ present. After time t_* at most $\mathsf{C}_{\mathsf{opt}} - t_*$ work in $\mathcal{A}$ can arrive because opt must be able to complete this work. Recalling that $\pi_{\mathcal{F}_{\mathsf{big}}{}'} \leq \mathsf{C}_{\mathsf{opt}}/\xi$, we have

$$\widetilde{\mathsf{C}}_{\mathsf{eve}} \leq t_* + \pi_{\mathcal{F}_{\mathsf{big}}{}'} + (\mathsf{C}_{\mathsf{opt}} - t_*) \leq \xi\mathsf{C}_{\mathsf{opt}}. \qquad \triangleleft$$

▷ Claim 14. $t_\star + \pi_{\mathcal{F}_{\mathsf{big}}{}'} + \pi_{\mathcal{A}_{\mathsf{early}}} < (\sigma_{\min} + R(\sigma_{\min}))/\xi$.

Proof. Fix a time $t < R(\sigma_{\min})$ after all tasks in $\mathcal{F}_{\mathsf{big}}{}' \cup \mathcal{A}_{\mathsf{early}}$ have arrived, and fix a task $\tau_i \in \mathcal{F}_{\mathsf{big}}{}'$ with $\sigma_i = \sigma_{\min}$. First, note that by Claim 13 we can assume that eve has not started τ_i by time t. Thus, eve is free to start τ_i on a slow machine, but chooses not to. This implies

$$\xi\mathsf{C}^t < \sigma_i + t < \sigma_{\min} + R(\sigma_{\min}). \tag{7}$$

We also observe that $\mathsf{opt}(t)$ must run $\mathcal{F}_{\mathsf{big}}{}' \cup \mathcal{A}_{\mathsf{early}}$ on the fast machine, since running any of them on the slow machine would finish after time $\sigma_{\min}$. Thus,

$$t_\star + \pi_{\mathcal{F}_{\mathsf{big}}{}'} + \pi_{\mathcal{A}_{\mathsf{early}}} \leq \mathsf{C}^t. \tag{8}$$

Combining Equation (8) and Equation (7) gives the desired statement. $\triangleleft$

The other observation we make is that $R(\sigma_{\min})$ cannot happen too early.

▷ Claim 15. $R(\sigma_{\min}) > (\xi - 1)\sigma_{\min}$.

Proof. Let $\tau_i \in \mathcal{F}_{\mathsf{big}}{}'$ be a task with $\sigma_i = \sigma_{\min}$. Note that $t_i \leq R(\sigma_{\min})$ by Claim 10. Then, by Claim 13 we may assume without loss of generality that at time $R(\sigma_{\min})$ eve is not running τ_i, and does not choose to start τ_i on a slow machine. So,

$$R(\sigma_{\min}) + \sigma_{\min} > \xi \mathsf{C}^{R(\sigma_{\min})} \geq \xi\sigma_{\min}.$$ ◁

Now we conclude the theorem.

▷ Claim 16. $\widetilde{\mathsf{C}}_{\mathsf{eve}} \leq \xi \mathsf{C}_{\mathsf{opt}}$.

Proof. Note that $\pi_{\mathcal{A}_{\text{late}}} \leq \mathsf{C}_{\mathsf{opt}} - R(\sigma_{\min})$. Also, note that since τ_s was not put on a slow machine immediately upon arrival, we must have $\pi_s \leq \sigma_s/\xi \leq \mathsf{C}_{\mathsf{opt}}/\xi^2$. Then, applying Claim 14 and Claim 15 we have

$$\begin{aligned}
\widetilde{\mathsf{C}}_{\mathsf{eve}} &\leq \pi_s + t_\star + \pi_{\mathcal{F}_{\mathsf{big}}{}'} + \pi_{\mathcal{A}_{\mathsf{early}}} + \pi_{\mathcal{A}_{\text{late}}} \\
&\leq \mathsf{C}_{\mathsf{opt}}/\xi^2 + (\sigma_{\min} + R(\sigma_{\min}))/\xi + \mathsf{C}_{\mathsf{opt}} - R(\sigma_{\min}) \\
&\leq (1/\xi^2 + (1 + \xi - 1)/\xi + 1 - (\xi - 1))\mathsf{C}_{\mathsf{opt}} \\
&= (3 + 1/\xi^2 - \xi)\mathsf{C}_{\mathsf{opt}} \\
&= \xi \mathsf{C}_{\mathsf{opt}}.
\end{aligned}$$ ◁

◀

▶ Remark 17. The simple nature of the lower bound in Proposition 32, along with the fact that eve gets a competitive ratio quite close to ϕ might leave the impression that ϕ is clearly the correct competitive ratio, and a slightly better analysis of (a natural variant of) eve would be ϕ-competitive. However, this is not the case: in Section A, we show that no **non-procrastinating** eventually-committing scheduler can achieve competitive ratio better than ξ, where a scheduler is called non-procrastinating if, whenever tasks are present, it always runs at least one task. Thus, if the competitive ratio of eve can be improved upon, doing so will require a substantially different scheduler: one which occasionally decides to do nothing at all despite work being present.

5 A 1.5-Competitive Never-Committing Scheduler

In this section we analyze never-committing schedulers. First we give a simple lower bound.

▶ **Proposition 18.** *Fix $\varepsilon > 0$. There is no deterministic $((1 + \sqrt{3})/2 - \varepsilon)$-competitive never-committing scheduler.*

Proof. We may assume $\varepsilon < .001$. Let $\psi = (\sqrt{3} - 1)/2$. Let $\mathcal{T} = \Big((1, 2\psi, 0), (\infty, 1 - \psi, \psi)\Big)$; that is, τ_1 has $\sigma_1 = 1, \pi_1 = 2\psi, t_1 = 0$, and τ_2 has $\sigma_2 = \infty, \pi_2 = 1 - \psi, t_2 = \psi$. Let $\mathcal{T}' = \Big((1, 2\psi, 0)\Big)$ be the same TAP without the second task. We have $\mathsf{C}^{\mathcal{T}'}_{\mathsf{opt}} = 2\psi$, since opt just runs the single task on the fast machine, and $\mathsf{C}^{\mathcal{T}}_{\mathsf{opt}} = 1$, since opt can run τ_1 on a slow machine and τ_2 on the fast machine.

Suppose that alg is a $(\psi + 1 - \varepsilon)$-competitive scheduler. On TAP $\mathcal{T}'$, at time $\psi - \varepsilon/2$, we claim alg must be running τ_1 on the fast machine. If not, then alg's completion time must be at least $\min(\sigma_1, \psi - \varepsilon/2 + \pi_1) = 1$, with the branch of the min depending on whether τ_1 is ever moved to the fast machine – but this gives competitive ratio $1/(2\psi) = 1 + \psi$.

Before time ψ, it is impossible to distinguish between $\mathcal{T}$ and $\mathcal{T}'$. Thus, alg must be running τ_1 on the fast machine at time $\psi - \varepsilon/2$ on TAP $\mathcal{T}$. Now, we have $\mathsf{C}^{\mathcal{T}}_{\mathsf{alg}} \geq \min(\sigma_1 + \psi - \varepsilon/2, \pi_1 + \pi_2) = 1 + \psi - \varepsilon/2$, with the branch of the min depending on whether τ_1 is ever moved to a slow machine – but this gives competitive ratio $(1 + \psi - \varepsilon/2)/1$. Thus, alg is not actually $(\psi + 1 - \varepsilon)$-competitive. ◀

Now we give a 1.5-competitive never-committing scheduler, which we call nev ("*nev*er committing"). Note that this competitive ratio is smaller than the lower bound of $\phi \approx 1.618$ known for eventually committing schedulers, so this demonstrates a separation between the strengths of schedulers in the two models.

▶ **Scheduler 19.** At time t:

- If task τ_i has $\sigma_i + t \leq 1.5\mathsf{C}^t$ but is not currently running on a slow machine, start τ_i on a slow machine, cancelling its fast machine implementation if necessary.
- Let $\mathcal{P}$ be the set of τ_i that have arrived and are not running on a slow machine. Choose $\tau_i \in \mathcal{P}$ maximizing $\sigma_i + t_i$, breaking ties by choosing the task with the smaller i. Run τ_i on the fast machine during this time step.

▶ **Theorem 20.** nev *is a* 1.5*-competitive never-committing scheduler.*

Proof. Fix TAP $\mathcal{T}$. Let $\widetilde{C}_{\mathsf{nev}}$ denote the final time when nev has work on the fast machine. Observe that if nev ever runs τ_i on a slow machine, then nev finishes τ_i before time $1.5\mathsf{C}_{\mathsf{opt}}$. Thus, to show that nev is 1.5-competitive it suffices to show $\widetilde{C}_{\mathsf{nev}} \leq 1.5\mathsf{C}_{\mathsf{opt}}$.

Let $\mathcal{A} = \{\tau_i : \sigma_i + t_i > \mathsf{C}_{\mathsf{opt}}\}$ be the set of tasks that opt *actually* runs on the fast machine.

▷ Claim 21. nev never runs a task $\tau \in \mathcal{A}$ on the fast machine after time $\mathsf{C}_{\mathsf{opt}}$.

Proof. nev always allocates the fast machine to the present task with the largest value of $\sigma_i + t_i$ among tasks that aren't running on slow machines. Thus, whenever there are tasks from $\mathcal{A}$ that aren't running on slow machines, nev will run one such task on the fast machine. opt is able to complete all tasks in $\mathcal{A}$ on the fast machine by time $\mathsf{C}_{\mathsf{opt}}$. Thus, nev completes or starts on slow machines all tasks $\tau \in \mathcal{A}$ before time $\mathsf{C}_{\mathsf{opt}}$. ◁

This means that the only way to have $\widetilde{\mathsf{C}}_{\mathsf{eve}} > \mathsf{C}_{\mathsf{opt}}$ is if there are tasks with $\sigma_i + t_i \leq \mathsf{C}_{\mathsf{opt}}$ that have yet to be completed at time $\mathsf{C}_{\mathsf{opt}}$; we assume that this is the case for the remainder of the proof. Let $\Pi(x)$ be the total amount of work nev performs on the fast machine after time $\mathsf{C}_{\mathsf{opt}}$ across all tasks with $\sigma_i + t_i \in [x, 1.5x]$. For any $x \leq \mathsf{C}_{\mathsf{opt}}$, let $R(x) = \inf\{t \mid \mathsf{C}^t \geq x\}$ be the first time an online algorithm becomes aware that the optimal schedule requires x completion time; the following key claim allows us to bound this left-over work $\Pi(x)$ in terms of $R(x)$.

▷ Claim 22. For all x, we have $\Pi(x) \leq x - R(x)$.

Proof. Let $\mathcal{J}_x$ denote the set of tasks with $\sigma_i + t_i \leq 1.5x$ that nev runs on the fast machine at some time after $\mathsf{C}_{\mathsf{opt}}$. First, note that all $\tau_i \in \mathcal{J}_x$ must have $t_i < R(x)$, or else τ_i would be placed on a slow machine upon arrival. Choose ε sufficiently small, such that no tasks arrive between times $R(x) - \varepsilon$ and $R(x)$. Since $R(x) - \varepsilon < R(x)$, we know $\mathsf{C}^{R(x)-\varepsilon} < x$, and so $\mathsf{opt}(R(x) - \varepsilon)$ must run all tasks with $\sigma_i + t_i \geq x$ on the fast machine. In order for $\mathsf{opt}(R(x) - \varepsilon)$ to finish these tasks before time $\mathsf{C}^{R(x)-\varepsilon} < x$, $\mathsf{opt}(R(x) - \varepsilon)$ must have at most $x - R(x) + \varepsilon$ fast work remaining across all such tasks.

Now, by the same argument as in Claim 21, because nev prioritizes tasks with $\sigma_i + t_i \geq x$ over tasks with $\sigma_i + t_i < x$ on the fast machine whenever they are present (and not yet started on slow machines), nev has at most $x - R(x) + \varepsilon$ work remaining on tasks in $\mathcal{J}_x$ at time $R(x) - \varepsilon$. Because no more tasks from $\mathcal{J}_x$ arrive after this time, we have $\Pi(x) \leq x - R(x) + \varepsilon$ as well. The claim held for all $\varepsilon > 0$, and so taking $\varepsilon \to 0$ we have $\Pi(x) \leq x - R(x)$. ◁

We now give an observation to control $R(x)$. Let $\tau_{i_\star}$ be the task, among all tasks that nev runs on the fast machine after time $\mathsf{C}_{\mathsf{opt}}$, with the smallest value of $\sigma_i + t_i$. Let $\lambda = \sigma_{i_\star} + t_{i_\star}$.

▷ Claim 23. For all $x \geq \lambda$, we have $R(x) > 1.5x - \lambda$.

Proof. First, note that $t_{i_\star} \leq R(\lambda) \leq \lambda$ or else nev would start $\tau_{i_\star}$ on a slow machine upon arrival. Now, because nev doesn't start $\tau_{i_\star}$ on a slow machine at time $R(x) > t_{i_\star}$, we have $R(x) + \sigma_{i_\star} > 1.5x$. ◁

To prove the theorem, it will now suffice to branch into two cases, based on how large λ is.

Case 1. $\lambda \geq (2/3)\mathsf{C}_{\mathsf{opt}}$.

In this case, since $1.5\lambda \geq \mathsf{C}_{\mathsf{opt}}$, by Claim 21 all left-over work at time $\mathsf{C}_{\mathsf{opt}}$ comes from tasks with $\sigma_i + t_i \in [\lambda, 1.5\lambda]$. By Claim 22, the total amount of such work is at most $\lambda - R(\lambda)$. Then, by Claim 23, we know $R(\lambda) > .5\lambda$. Together, this implies that the total amount of leftover work is at most $.5\lambda \leq .5\mathsf{C}_{\mathsf{opt}}$.

Case 2. $\lambda < 2\mathsf{C}_{\mathsf{opt}}/3$.

First note that $\lambda \geq \mathsf{C}_{\mathsf{opt}}/2$ or else $\tau_{i_\star}$ would be started on a slow machine at time $\mathsf{C}_{\mathsf{opt}}$. So, all left-over work at time $\mathsf{C}_{\mathsf{opt}}$ comes either from tasks with $\sigma_i + t_i \in [\lambda, 1.5\lambda]$, or from tasks with $\sigma_i + t_i \in [1.5\lambda, 1.5^2\lambda]$. By Claim 22, we can therefore bound the total amount of leftover work by $(\lambda - R(\lambda)) + (1.5\lambda - R(1.5\lambda))$. Now, by Claim 23, this quantity can be at most $(\lambda - .5\lambda) + (1.5\lambda - 1.25\lambda) = .75\lambda$. Since $\lambda < 2\mathsf{C}_{\mathsf{opt}}/3$, this is at most $.5\mathsf{C}_{\mathsf{opt}}$. ◀

▶ Remark 24. In Section A we show that Scheduler 19 is optimal among never-committing schedulers that never cancel implementations running on slow machines. This shows that improving on Scheduler 19 will require a substantially different scheduler.

6 Extending Beyond the Massively Parallel Regime

In Theorem 7, we have shown that Scheduler 5 is a 2-competitive instantly-committing scheduler in the Massively Parallel regime of the SPDP. In this section, we will show that in fact, Scheduler 5 is a 2-competitive scheduler even outside of the Massively Parallel regime, although the analysis is slightly more complicated. This result is interesting in its own right, resolving an open question from [16]. However, we think that the main virtue of this proof is that it serves as a proof-of-concept that results from the conceptually simpler Massively Parallel regime can be adapted to apply to the general SPDP: we conjecture that all upper bounds holding in the massively parallel regime should also hold in the general SPDP.

First, recall the setup of the SPDP problem. The input is a sequence of triples $\tau_i = (\sigma_i, \pi_i, t_i)$, where π_i is the work of the parallel implementation of task τ_i, and σ_i is the work of the serial implementation of task τ_i. At each time step, the scheduler allocates its processors to the jobs, giving at most 1 processor to each serial job.

Scheduler 5 is not a defined scheduler in the SPDP, because we specify the decisions for which tasks to run, but do not specify how to schedule the tasks. We extend ins to the general SPDP as follows:

▶ **Scheduler 25.** When task τ_i arrives:
- If $\sigma_i + t_i > 2\mathsf{C}^{t_i}$ parallelize τ_i.
- Otherwise, serialize τ_i.

At every timestep, if there are x serial jobs present, then ins schedules the jobs by allocating a processor to each of the $\min(p, x)$ serial jobs with the most remaining work (or an arbitrary set of p jobs with maximum remaining work if there are more than p serial jobs with maximum remaining work), and then allocating any remaining processors to an arbitrary parallel job (if a parallel job is present).

Now we analyze ins. We say ins is **saturated** at time t if ins has no idle processors at time t.

▶ **Lemma 26.** *If* ins *is unsaturated right before finishing, then* $\mathsf{C}_{\mathsf{ins}} \leq 2\mathsf{C}_{\mathsf{opt}}$.

Proof. We claim that if ins is unsaturated at time t, then for each task τ_i present at time t, τ_i has been run on every time step since it arrived. Suppose this is not the case. Then, there must have been some time step before time t when there were at least p serial jobs with at least as much remaining work as τ_i. But then τ_i will finish at the same time as these other jobs, contradicting the fact that ins is unsaturated at time t. Thus, if ins is unsaturated at time t, then $t \leq \sigma_i + t_i$ for some i such that ins ran τ_i in serial. Thus, $t \leq 2\mathsf{C}_{\mathsf{opt}}$, as desired. ◀

By virtue of Lemma 26 it suffices to consider the case that ins is saturated immediately before finishing. Let t_* be the final time in $[0, \mathsf{C}_{\mathsf{ins}})$ when ins is unsaturated (we set $t_* = 0$ if ins is always saturated). Let $i_* \in [n]$ be the smallest i such that $t_i \geq t_*$; in fact we will have $t_{i_*} = t_*$, since in order to transition from being unsaturated to being saturated, some tasks must arrive. For integer $i \in [i_*, n]$, let $\mathsf{K}^i_{\mathsf{alg}}$ denote the sum of π_j for each τ_j with $i_* \leq j \leq i$ that alg runs in parallel; If alg is an instantly-committing scheduler then $\mathsf{K}^i_{\mathsf{alg}}$ can be computed at time t_i. Now we prove an analogue of Lemma 6.

▶ **Lemma 27.** *Fix a length n TAP. For all $i \in [i_*, n]$, and for all instantly-committing schedulers* alg*,*

$$\mathsf{K}^i_{\mathsf{ins}} \leq (\mathsf{C}^{t_i}_{\mathsf{alg}} - t_*)p + \mathsf{K}^i_{\mathsf{alg}}. \tag{9}$$

Proof. We prove Equation (9) by induction on i. The base case is $i = i_*$. If alg takes π_i work here, then we have $\mathsf{K}^i_{\mathsf{ins}} \leq \mathsf{K}^i_{\mathsf{alg}}$ (and $\mathsf{C}^{t_i}_{\mathsf{alg}} \geq t_*$) so the invariant will hold. If instead alg runs τ_i in serial, then we'll have $\mathsf{C}^{t_i}_{\mathsf{alg}} - t_* \geq \sigma_i \geq \pi_i/p$, in which case we have $\mathsf{K}^i_{\mathsf{ins}} \leq (\mathsf{C}^{t_i}_{\mathsf{alg}} - t_*)p$ so the invariant holds. This establishes the base case.

Now, assume Equation (9) for $i \in [i_*, n)$; we prove Equation (9) for $i + 1$. If alg takes at least as much work as ins on τ_{i+1}, i.e., $\mathsf{K}^{i+1}_{\mathsf{alg}} - \mathsf{K}^i_{\mathsf{alg}} \geq \mathsf{K}^{i+1}_{\mathsf{ins}} - \mathsf{K}^i_{\mathsf{ins}}$, then Equation (9) for (i, alg) implies Equation (9) for $(i + 1, \mathsf{alg})$. It remains to consider the case when alg runs τ_{i+1} in serial, but ins runs τ_{i+1} in parallel. Here, ins thought τ_{i+1} was too large to serialize large, so:

$$\mathsf{C}^{t_{i+1}}_{\mathsf{alg}} \geq \sigma_{i+1} + t_{i+1} > 2\mathsf{C}^{t_{i+1}}. \tag{10}$$

One consequence of Equation (10) is that $\mathsf{opt}(t_{i+1})$ parallelizes τ_{i+1}; a corollary of this is that Equation (9) holds for $(i+1, \mathsf{opt}(t_{i+1}))$. Thus, applying Equation (9) for $(i+1, \mathsf{opt}(t_{i+1}))$ and using Equation (10) gives:

$$\mathsf{K}_{\mathsf{ins}}^{i+1} \leq (\mathsf{C}^{t_{i+1}} - t_*)p + \mathsf{K}_{\mathsf{opt}(t_{i+1})}^{i+1} \leq (2\mathsf{C}^{t_{i+1}} - 2t_*)p \leq (\mathsf{C}_{\mathsf{alg}}^{t_{i+1}} - t_*)p,$$

so the invariant holds in this case as well. ◀

In the Massively Parallel regime Theorem 7 followed immediately from Lemma 6. Slightly more work is required in the general setting, but Lemma 27 is still very useful.

▶ **Theorem 28.** ins *is a 2-competitive instantly-committing scheduler in the SPDP.*

Proof. Recall from Lemma 26 that we need only consider the case that ins ends saturated, and recall the definition of t_*. For any scheduler alg, let $\mathsf{B}_{\mathsf{alg}}$ denote the work that alg has left immediately before time t_*, and let $\mathsf{K}_{\mathsf{alg}}$ be work that alg takes on tasks τ_i with $t_i \geq t_*$. Because ins ends saturated, we have

$$\mathsf{C}_{\mathsf{ins}} = t_* + (\mathsf{K}_{\mathsf{ins}} + \mathsf{B}_{\mathsf{ins}})/p.$$

Applying Lemma 27 gives

$$t_* + (\mathsf{K}_{\mathsf{ins}} + \mathsf{B}_{\mathsf{ins}})/p \leq \mathsf{C}_{\mathsf{opt}} + (\mathsf{K}_{\mathsf{opt}} + \mathsf{B}_{\mathsf{ins}})/p. \tag{11}$$

So, to conclude, it suffices to show that $\mathsf{B}_{\mathsf{ins}} + \mathsf{K}_{\mathsf{opt}} \leq p\mathsf{C}_{\mathsf{opt}}$. Let S be the set of tasks that ins has present immediately before time t_*. Let $W = \sum_{\tau_i \in S} \sigma_i$. Clearly $\mathsf{B}_{\mathsf{ins}} \leq W$. On the other hand, opt must take at least W work on the tasks S, and can have made at most pt_* progress on these tasks by time t_*. Thus,

$$\mathsf{B}_{\mathsf{ins}} \leq W \leq pt_* + \mathsf{B}_{\mathsf{opt}}.$$

Therefore,

$$\mathsf{B}_{\mathsf{ins}} + \mathsf{K}_{\mathsf{opt}} \leq pt_* + \mathsf{B}_{\mathsf{opt}} + \mathsf{K}_{\mathsf{opt}} \leq p\mathsf{C}_{\mathsf{opt}}.$$

Using this in Equation (11) gives $\mathsf{C}_{\mathsf{ins}} \leq 2\mathsf{C}_{\mathsf{opt}}$. ◀

References

1. S Anand, Naveen Garg, and Nicole Megow. Meeting deadlines: How much speed suffices? In *Automata, Languages and Programming: 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part I 38*, pages 232–243. Springer, 2011. doi:10.1007/978-3-642-22006-7_20.
2. James Aspnes, Yossi Azar, Amos Fiat, Serge Plotkin, and Orli Waarts. On-line routing of virtual circuits with applications to load balancing and machine scheduling. *Journal of the ACM (JACM)*, 44(3):486–504, 1997. doi:10.1145/258128.258201.
3. Baruch Awerbuch, Yossi Azar, Edward F Grove, Ming-Yang Kao, P Krishnan, and Jeffrey Scott Vitter. Load balancing in the l_p norm. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 383–391. IEEE, 1995. doi:10.1109/SFCS.1995.492494.
4. Brenda S Baker and Jerald S Schwarz. Shelf algorithms for two-dimensional packing problems. *SIAM Journal on Computing*, 12(3):508–525, 1983. doi:10.1137/0212033.
5. Ioannis Caragiannis. Better bounds for online load balancing on unrelated machines. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 972–981, 2008. URL: http://dl.acm.org/citation.cfm?id=1347082.1347188.

6 Shichuan Deng, Jian Li, and Yuval Rabani. Generalized unrelated machine scheduling problem. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2898–2916. SIAM, 2023. doi:10.1137/1.9781611977554.CH110.

7 Richard A. Dutton and Weizhen Mao. Online scheduling of malleable parallel jobs. In *Proceedings of the 19th IASTED International Conference on Parallel and Distributed Computing and Systems*, PDCS '07, pages 136–141, USA, 2007. ACTA Press.

8 R. L. Graham. Bounds on multiprocessing timing anomalies. *SIAM Journal on Applied Mathematics*, 17(2):416–429, 1969. doi:10.1137/0117039.

9 Shouwei Guo and Liying Kang. Online scheduling of malleable parallel jobs with setup times on two identical machines. *European Journal of Operational Research*, 206(3):555–561, November 2010. doi:10.1016/j.ejor.2010.03.005.

10 Anupam Gupta, Amit Kumar, Viswanath Nagarajan, and Xiangkun Shen. Stochastic load balancing on unrelated machines. *Mathematics of Operations Research*, 46(1):115–133, 2021. doi:10.1287/MOOR.2019.1049.

11 Varun Gupta, Benjamin Moseley, Marc Uetz, and Qiaomin Xie. Stochastic online scheduling on unrelated machines. In *Integer Programming and Combinatorial Optimization: 19th International Conference, IPCO 2017, Waterloo, ON, Canada, June 26-28, 2017, Proceedings 19*, pages 228–240. Springer, 2017. doi:10.1007/978-3-319-59250-3_19.

12 Varun Gupta, Benjamin Moseley, Marc Uetz, and Qiaomin Xie. Greed works—online algorithms for unrelated machine stochastic scheduling. *Mathematics of operations research*, 45(2):497–516, 2020. doi:10.1287/MOOR.2019.0999.

13 Ellis Horowitz and Sartaj Sahni. Exact and approximate algorithms for scheduling nonidentical processors. *J. ACM*, 23(2):317–327, April 1976. doi:10.1145/321941.321951.

14 Johann L Hurink and Jacob Jan Paulus. Online algorithm for parallel job scheduling and strip packing. In *Approximation and Online Algorithms: 5th International Workshop, WAOA 2007, Eilat, Israel, October 11-12, 2007. Revised Papers 5*, pages 67–74. Springer, 2008. doi:10.1007/978-3-540-77918-6_6.

15 Sungjin Im and Shi Li. Improved approximations for unrelated machine scheduling. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2917–2946. SIAM, 2023. doi:10.1137/1.9781611977554.CH111.

16 William Kuszmaul and Alek Westover. Scheduling jobs with work-inefficient parallel solutions. In *Proceedings of the 36th ACM Symposium on Parallelism in Algorithms and Architectures*, pages 101–111, 2024. doi:10.1145/3626183.3659960.

17 Jan Karel Lenstra, David B Shmoys, and Éva Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical programming*, 46:259–271, 1990. doi:10.1007/BF01585745.

18 Walter Ludwig and Prasoon Tiwari. Scheduling malleable and nonmalleable parallel tasks. In *Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 167–176, 1994. URL: http://dl.acm.org/citation.cfm?id=314464.314491.

19 Marco Molinaro. *Stochastic ℓ_p Load Balancing and Moment Problems via the L-Function Method*, pages 343–354. SIAM, 2019. doi:10.1137/1.9781611975482.22.

20 Gregory Mounie, Christophe Rapine, and Dennis Trystram. Efficient approximation algorithms for scheduling malleable tasks. In *Proceedings of the eleventh annual ACM symposium on Parallel algorithms and architectures*, SPAA '99, pages 23–32, New York, NY, USA, 1999. Association for Computing Machinery. doi:10.1145/305619.305622.

21 Daniel R. Page, Roberto Solis-Oba, and Marten Maack. Makespan minimization on unrelated parallel machines with simple job-intersection structure and bounded job assignments. *Theoretical Computer Science*, 809:204–217, 2020. doi:10.1016/j.tcs.2019.12.009.

22 Andreas S. Schulz and Martin Skutella. Scheduling unrelated machines by randomized rounding. *SIAM Journal on Discrete Mathematics*, 15(4):450–469, 2002. doi:10.1137/S0895480199357078.

23 John Turek, Walter Ludwig, Joel L. Wolf, Lisa Fleischer, Prasoon Tiwari, Jason Glasgow, Uwe Schwiegelshohn, and Philip S. Yu. Scheduling parallelizable tasks to minimize average response time. In *Proceedings of the sixth annual ACM symposium on Parallel algorithms and architectures*, SPAA '94, pages 200–209, New York, NY, USA, 1994. Association for Computing Machinery. doi:10.1145/181014.181331.

24 John Turek, Uwe Schwiegelshohn, Joel L Wolf, and Philip S Yu. Scheduling parallel tasks to minimize average response time. In *Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 112–121, 1994. URL: http://dl.acm.org/citation.cfm?id=314464.314485.

25 John Turek, Joel L Wolf, and Philip S Yu. Approximate algorithms scheduling parallelizable tasks. In *Proceedings of the fourth annual ACM symposium on Parallel algorithms and architectures*, pages 323–332, 1992. doi:10.1145/140901.141909.

26 Nodari Vakhania, Jose Hernandez, and Frank Werner. Scheduling unrelated machines with two types of jobs. *International Journal of Production Research*, 52:1–9, February 2014. doi:10.1080/00207543.2014.888789.

27 Andrew Chi-Chin Yao. Probabilistic computations: Toward a unified measure of complexity. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 222–227. IEEE Computer Society, 1977.

28 Deshi Ye, Danny Z. Chen, and Guochuan Zhang. Online scheduling of moldable parallel tasks. *Journal of Scheduling*, 21(6):647–654, 2018. Publisher: Springer. URL: https://ideas.repec.org//a/spr/jsched/v21y2018i6d10.1007_s10951-018-0556-2.html, doi:10.1007/S10951-018-0556-2.

29 Deshi Ye, Xin Han, and Guochuan Zhang. A note on online strip packing. *Journal of Combinatorial Optimization*, 17(4):417–423, 2009. doi:10.1007/S10878-007-9125-X.

30 Xiaoyan Zhang, Ran Ma, Jian Sun, and Zan-Bo Zhang. Randomized selection algorithm for online stochastic unrelated machines scheduling. *Journal of Combinatorial Optimization*, pages 1–16, 2022.

A Barriers Against Improved Schedulers

In this section we show that the schedulers of Section 4 and Section 5 are optimal among natural restricted classes of schedulers. This highlights what changes must be made to the schedulers in order to have hopes of achieving better competitive ratios.

First we show that among **non-procrastinating** eventually-committing schedulers (i.e., eventually-committing schedulers with the property that whenever tasks are present, they will run at least one task), the scheduler Scheduler 8 is optimal.

▶ **Proposition 29.** *Fix $\varepsilon > 0$. Let $\xi \approx 1.677$ denote the real root of the polynomial $2x^3 - 3x^2 - 1$. There is no deterministic $(\xi - \varepsilon)$-competitive non-procrastinating eventually-committing scheduler.*

Proof. It suffices to consider the case that $\varepsilon < .001$. Fix a non-procrastinating eventually-committing scheduler alg. Assume towards contradiction that alg is $(\xi - \varepsilon)$-competitive. We now describe a TAP $\mathcal{T}$ on which $\mathsf{C}_{\mathsf{alg}} \geq \xi \mathsf{C}_{\mathsf{opt}}$. The TAP starts with $\tau_1 = (1/\xi, 1/\xi^2, 0)$. Next, let $\tau_2 = (1 - \varepsilon^2, 1/\xi - \varepsilon^2, \varepsilon^2)$. Finally, at each time $t \in [\xi + 1/\xi - 2, 1 - \varepsilon^2] \cap (\mathbb{N}\varepsilon^2)$, give a task $\tau = (\infty, \varepsilon^2, t)$.

We now argue that alg must run all the tasks on the fast machine. Because alg is a non-procrastinating $(\xi - \varepsilon)$-competitive scheduler, alg must instantly start τ_1 on the fast machine (in case there are no tasks after τ_1). Now we argue that alg runs τ_2 on the fast machine as well. Suppose that alg starts τ_2 on a slow machine at some time t with

$$1 - \varepsilon^2 + t > (\xi - \varepsilon)\mathsf{C}^t. \tag{12}$$

Then, alg would not be $(\xi - \varepsilon)$-competitive on the truncated TAP $\mathcal{T}^t$. Thus, alg must not start τ_2 on a slow machine at any time t satisfying Equation (12). We now show that Equation (12) holds for all $t \geq \varepsilon$, thus proving that alg must run τ_2 on the fast machine. For $t \in [\varepsilon^2, \xi + 1/\xi - 2)$ we have $\mathsf{C}^t \leq 1/\xi$, and $1 - \varepsilon^2 + t \geq 1$, so Equation (12) holds. For $t \geq \xi + 1/\xi - 2$ we have

$$\mathsf{C}^t \leq \min(1, t - \xi + 2 + \varepsilon^2).$$

Thus, it suffices to show:

$$1 - \varepsilon^2 + t > (\xi - \varepsilon)\min(1, t - \xi + 2 + \varepsilon^2). \tag{13}$$

To show Equation (13) it suffices to check Equation (13) for $t = \xi - 1 - \varepsilon^2$ (by monotonicity of the inequality on either side of $t = \xi - 1$). At $t = \xi - 1 - \varepsilon^2$ Equation (13) is:

$$\xi - 2\varepsilon^2 > \xi - \varepsilon,$$

which is true because $\varepsilon < .001$.

We have now shown that alg runs all tasks in $\mathcal{T}$ on the fast machine. Thus, (by definition of ξ)

$$\mathsf{C}_{\mathsf{alg}} \geq 1/\xi^2 + 1/\xi - \varepsilon^2 + 1 - (1/\xi + \xi - 2) - \varepsilon^2 = \xi - 2\varepsilon^2.$$

However, $\mathsf{C}_{\mathsf{opt}} \leq 1$. This contradicts the assumption that alg is $(\xi - \varepsilon)$-competitive. ◀

Now, we show that the 1.5-competitive scheduler of Section 5 is optimal among never-committing schedulers that don't cancel tasks on slow machines.

▶ **Proposition 30.** *Let* alg *be a deterministic never-committing scheduler that never cancels serial tasks. Then, for any $\varepsilon > 0$, there is a TAP $\mathcal{T}$ with $n \leq O(1)$ on which* alg *has is not $(1.5 - \varepsilon)$-competitive.*

Proof. It suffices to consider the case that $\varepsilon < .001$. The TAP is defined as follows. First, $\tau_1 = (2, 1, 0)$. Then, for each time $t \in [\varepsilon^2, 1 - \varepsilon^2] \cap \mathbb{N}\varepsilon^2$, a task $\tau = (\infty, \varepsilon^2, t)$ arrives. We will show that if alg starts τ_1 on a slow machine at any time t then alg is not $(1.5 - \varepsilon)$-competitive on $\mathcal{T}^t$. We show this by considering two cases.

Case 1. alg starts τ_1 on a slow machine at time $t \in [0, 1]$.
If alg does this, then $\mathsf{C}^{\mathcal{T}^t}_{\mathsf{alg}} \geq 2 + t$. However, $\mathsf{C}^t \leq t + 1 + \varepsilon^2$. Thus,

$$\mathsf{C}^{\mathcal{T}^t}_{\mathsf{alg}}/\mathsf{C}^t \geq \frac{2 + t}{t + 1 + \varepsilon^2} \geq \frac{3}{2 + \varepsilon^2} > 1.5 - \varepsilon.$$

So alg cannot start τ_1 on a slow machine at this time.

Case 2. alg starts τ_1 on a slow machine at time $t \geq 1$.
If alg does this, then $\mathsf{C}_{\mathsf{alg}} \geq 2 + t$. However, $\mathsf{C}_{\mathsf{opt}} \leq 2$. Thus,

$$\mathsf{C}_{\mathsf{alg}}/\mathsf{C}_{\mathsf{opt}} \geq 1.5.$$

In conclusion, alg must run τ_1 on the fast machine. But then

$$\mathsf{C}_{\mathsf{alg}} \geq 3 - \varepsilon^2 > (1.5 - \varepsilon)\mathsf{C}_{\mathsf{opt}} = (1.5 - \varepsilon)2,$$

a contradiction. ◀

B Lower Bounds from [16]

In this section we state, for the reader's convenience, the lower bounds from [16] against instantly- and eventually- committing schedulers.

▶ **Proposition 31** (Kuszmaul, Westover [16]). *Fix $\varepsilon > 0$. There is no deterministic $(2-\varepsilon)$-competitive instantly-committing scheduler.*

Proof. Consider an n-task TAP where for each $i \in [n]$, the i-th task has $\sigma_i = 2^i, \pi_i = 2^{i-1}$, and the arrival times are all very close to 0. For each $i \in [n]$, it is possible to handle the first i tasks in the TAP with completion time 2^{i-1}. Thus, a $(2-\varepsilon)$-competitive scheduler cannot afford to run task τ_i on a slow machine. So, a $(2-\varepsilon)$-competitive scheduler must run all tasks on the fast machine, giving completion time at least $2^n - 1$ on this TAP, while $\mathsf{C}_{\mathsf{opt}} \leq 2^{n-1}$. For large enough n this implies that the scheduler is not actually $2-\varepsilon$ competitive. ◀

▶ **Proposition 32** (Kuszmaul, Westover [16]). *Fix $\varepsilon > 0$. There is no deterministic $(\phi-\varepsilon)$-competitive eventually-committing scheduler, where $\phi \approx 1.618$ is the golden ratio.*

Proof. Suppose that alg is a $(\phi-\varepsilon)$-competitive eventually-committing scheduler. Let $\tau_1 = (\phi, 1, 0)$; if there are no further tasks, alg must run τ_1 on the fast machine, starting at some time $t_0 \leq 1/\phi$. Let $\tau_2 = (\infty, \phi - t_0, t_0)$. On this TAP, $\mathsf{C}_{\mathsf{opt}} = \phi$, while $\mathsf{C}_{\mathsf{alg}} \geq \phi + 1 = \phi^2$. So alg is not $(\phi-\varepsilon)$-competitive. ◀

C Randomized Lower Bounds

In this section we give lower bounds against randomized schedulers. Our main tool is Yao's minimax principle [27], which allows us to prove a lower bound on the competitive ratio by exhibiting a distribution over TAPs, and showing that any deterministic scheduler has poor expected cost on a random TAP drawn from the distribution.

▶ **Proposition 33.** *For any $\varepsilon > 0.03$, there is no $(5/3-\varepsilon)$-competitive instantly-committing scheduler, even with randomization.*

Proof. Fix $N = 25$. For $k \in \mathbb{N}$, define $\mathcal{T}_k$ to be a length k TAP with $\sigma_i = 2^{i+1}, \pi_i = 2^i$. Let $\mathcal{D}$ denote the following distribution over TAPs: choose $k \in [N]$ uniformly randomly, and then output TAP $\mathcal{T}_k$. By brute force enumeration of all possible deterministic instantly-committing strategies, one can show that no such strategy is 1.637-competitive on this TAP. ◀

▶ **Proposition 34.** *For any $\varepsilon > 0$, there is no $((3+\sqrt{3})/4-\varepsilon)$-competitive eventually-committing scheduler, even with randomization.*

Proof. In the proof of Proposition 18 we defined two TAPs, and showed that no deterministic eventually-committing scheduler is $((1+\sqrt{3})/2-\varepsilon)$-competitive on both of the TAPs. One can show that if we randomly choose between the two TAPs of Proposition 18, there is no deterministic eventually-committing scheduler with expected competitive ratio $(1+\sqrt{3})/4 - \varepsilon$. ◀

Detecting and Correcting Computationally Bounded Errors: A Simple Construction Under Minimal Assumptions

Jad Silbak ✉
Northeastern University, Boston, MA, USA

Daniel Wichs ✉
Northeastern University, Boston, MA, USA
NTT Research, Sunnyvale, CA, USA

Abstract

We study error detection and error correction in a computationally bounded world, where errors are introduced by an arbitrary polynomial time adversarial channel. We consider codes where the encoding procedure uses random coins and define two distinct variants: (1) in *randomized codes*, fresh randomness is chosen during each encoding operation and is unknown a priori, while (2) in *self-seeded codes*, the randomness of the encoding procedure is fixed once upfront and is known to the adversary. In both cases, the randomness need not be known to the decoding procedure, and there is no trusted common setup between the encoder and decoder. The encoding and decoding algorithms are efficient and run in some fixed polynomial time, independent of the run time of the adversary.

The parameters of standard codes for worst-case (inefficient) errors are limited by the Singleton bound: for rate R it is not possible to detect more than a $1 - R$ fraction of errors, or uniquely correct more than a $(1 - R)/2$ fraction of errors, and efficient codes matching this bound exist for sufficiently large alphabets. In the computationally bounded setting, we show that going beyond the Singleton bound implies *one-way functions* in the case of randomized codes and *collision-resistant hash functions* in the case of self-seeded codes. We construct randomized and self-seeded codes under these respective minimal assumptions with essentially optimal parameters over a constant-sized alphabet:

- Detection: the codes have a rate $R \approx 1$ while detecting a $\rho \approx 1$ fraction of errors.
- Correction: for any $\rho < 1/2$, the codes uniquely correct a ρ fraction of errors with rate $R \approx 1 - \rho$.

Codes for computationally bounded errors were studied in several prior works starting with Lipton (STACS '94), but all such works either: (a) need some trusted common setup (e.g., public-key infrastructure, common reference string) between the encoder and decoder, or (b) only handle channels whose complexity is a prior bounded below that of the code.

2012 ACM Subject Classification Theory of computation → Cryptographic primitives

Keywords and phrases Error Correction, One-Way Functions, Collision Resistant Hashing

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.88

Related Version *Full Version*: `https://eprint.iacr.org/2024/1461` [22]

Funding *Jad Silbak*: Research supported by the Khoury College Distinguished Postdoctoral Fellowship.
Daniel Wichs: Research supported by NSF CNS-2349972 and CNS-2055510.

Acknowledgements We are very grateful to Ronen Shaltiel and Tal Yankovitz for very useful discussions.


© Jad Silbak and Daniel Wichs;
licensed under Creative Commons License CC-BY 4.0



1 Introduction

The theory of error detection and error correction goes back to the seminal works of Shannon [21] and Hamming [7]. A sender *encodes* a message and sends the resulting codeword over a noisy channel that may introduce errors by modifying a subset of the codeword symbols. A receiver gets the modified codeword and attempts to *decode* the original message. The goal of *error-detection* is to recover the original message when there are no errors or detect the presence of errors. The goal of *error-correction* is to always uniquely recover the original message. The main parameters of interest are the *rate* R, defined as the ratio of message length to codeword length, and the *relative error tolerance* ρ, defined as the ratio between the number of errors that can be detected/corrected and the codeword length.

Starting with the work of Hamming [7], a large body of research studies codes for worst-case computationally unbounded (adversarial) channels that can introduce arbitrary errors patterns. The error tolerance in this setting is fully determined by the *distance* of the code, which is subject to several well-established fundamental bounds. For example, the *Singleton bound* implies that for any rate $R \geq 0$ the relative error tolerance is $\rho \leq 1 - R$ for error detection and $\rho \leq (1 - R)/2$ for error correction. Furthermore, there are efficient codes matching the Singleton bound for sufficiently large alphabet sizes.

Computationally Bounded Channels

We consider a setting similar to that of Hamming [7], where errors are introduced by a fully adversarial channel, but with the additional natural restriction that the channel is *computationally bounded*, and can run in arbitrary polynomial (or even sub-exponential) time. This restriction is well motivated – if an error pattern takes longer than the lifetime of the universe to compute, then we can safely assume it will not occur. Our goal is to construct efficient codes that can detect/correct errors in this setting with rate vs error-tolerance trade-offs that go beyond the Singleton bound. Many prior works starting with [11] and including [14, 12, 8, 4, 17, 1, 18, 19, 20, 3] have previously considered computationally bounded channels, either to get better parameters, or to achieve other desirable properties such as improved local/list decoding. However, all such prior works suffer from at least one of the following deficiencies:

1. *Trusted setup:* Many works require a trusted coordinated setup between the encoding and decoding algorithms. For example, [11] assumes that they share a secret key, which is unknown to the adversarial channel. Alternatively, [12] require a public-key infrastructure where the encoding algorithm has its own secret key, while the decoding algorithm (as well as the adversarial channel) gets the corresponding public-key. In both cases, the encoding algorithm is also stateful and updates its state between operations. The use of secret keys and state are significant limitations that make such codes inapplicable in many natural settings. The work of Grossman, Holmgren and Yogev [3] introduces the notion of *seeded codes* that only requires a light trusted setup consisting of a *common (uniformly) random seed* shared by the encoder and decoder. They construct such seeded codes correcting $\rho \approx 1 - R$ errors over a large alphabet based on strong and non-standard cryptographic hardness assumptions – namely a certain form of *two-input correlation-intractable hash functions*, which are currently only known to exist in the random-oracle model, but have no known provably secure instantiations under standard cryptographic hardness assumptions.
2. *Codes have higher complexity than adversary:* Many works [4, 17, 18, 19, 20] restrict the complexity (time/size or space) of the adversarial channel to be bounded by some arbitrary polynomial chosen a priori, and the complexity of the encoding/decoding algorithms then

scales accordingly to some sufficiently larger polynomial. In particular, the adversarial channel is not powerful enough to evaluate the encoding/decoding algorithms in these schemes. This is a significant limitation, since we cannot argue that such schemes protect against all errors that can arise in a computationally bounded world – a world that can compute the encoding/decoding algorithms of the scheme can also find error patterns to defeat them! In contrast, we would like to have efficient codes with some fixed polynomial run-time that protect against arbitrary polynomial (or even sub-exponential time) adversarial channels.

The different prior works and their properties are summarized in Table 1.

Table 1 Summary of related work. Seeded codes are also known as Monte-Carlo codes in the literature and require a common random seed shared by the encoder and decoder as a trusted setup. Randomized codes require fresh on-the-fly randomness at each encoding operation. Seeded + randomized codes require both simultaneously. Codes for channels in $\text{SIZE}(n^c)$ have correspondingly larger run-time $\text{poly}(n^c)$, while codes for channels in P/poly have some a priori fixed polynomial run-time independent of the channel. Some works focus on the binary alphabet while others focus on larger alphabets.

work	Setup/Type	Channel	Decoding	Assumption
[11]	symmetric-key stateful encoding	P/poly	unique	one-way functions
[12]	public key stateful encoding	P/poly	unique	one-way functions
[4]	seeded + randomized	$\text{SIZE}(n^c)$	list	none
[17]	randomized	$\text{SIZE}(n^c)$	list	derandomization assumption
[18]	randomized	$\text{SPACE}(n^\delta)$	unique	none
[19]	seeded + randomized	$\text{SIZE}(n^c)$	unique	none
[20]	randomized	$\text{SIZE}(n^c)$	unique	derandomization assumption
[3]	seeded	P/poly	unique	correlation-intractable hash (random oracle)
This work	randomized	P/poly	unique	one-way functions
This work	self-seeded	P/poly	unique	collision-resistant hash

1.1 Our Results

In this work we construct error detection and error correction codes for computationally bounded channels, while simultaneously achieving: (1) no trusted setup, (2) a fixed polynomial-time code that protects against arbitrary polynomial (or even sub-exponential) time adversarial channels, (3) provable guarantees under basic/minimal computational hardness assumptions, (4) essentially optimal trade-offs between rate and error tolerance beyond the Singleton bound for sufficiently large constant-sized alphabets.

Randomized/Self-Seeded Codes

Standard deterministic codes cannot achieve better parameters against (non-uniform) computationally bounded channels beyond what they achieve for computationally unbounded channels, and in particular, they cannot go beyond the Singleton bound. This is because a worst-case message and error pattern for which detection/correction fails can simply be hard-coded as non-uniform advice to an efficient adversary. To overcome this limitation, we therefore consider codes (Enc, Dec), where the encoding procedure Enc takes additional

randomness as input. We consider two variants. In the basic variant of *randomized codes*, fresh randomness r for the encoding algorithm is chosen each time on the fly and is unknown to the adversary ahead of time. The adversary chooses a worst-case message m, gets the resulting randomized codeword $c \leftarrow \mathrm{Enc}(m; r)$ for a fresh random r, and then chooses the error e as an arbitrary efficiently computable function of the codeword. We also consider a stronger variant called *self-seeded codes*, where the randomness r of the encoding algorithm is fixed once and for all and known to the adversary a priori. That is, the adversary gets the randomness r and then uses it to adaptively choose the message m (which fully defines the codeword $c = \mathrm{Enc}(m; r)$) along with the error e. In both cases, the fractional Hamming weight of e is bounded by the error tolerance ρ. Error detection ensures that with overwhelming probability the resulting erroneous codeword $c' = c + e$ decodes to m when there are no errors, and decodes to either m or $\perp$ when there are errors. Error correction ensures that with overwhelming probability c' decodes to m.

The main technical difference between randomized and self-seeded codes is adaptivity: in the former the message m to be encoded is chosen by the adversary before the randomness r is chosen, while in the latter the message m can adaptively depend on r.[1] Self-seeded codes allow us to fix the randomness for the code once and for all as a public seed, and afterwards the code is fully deterministic. Moreover, the seed can be chosen by the encoder unilaterally without any coordination with the decoder. For example, different software manufacturers can individually choose their own seeds to be used by the encoding algorithms in their software, while allowing for interoperability with the decoders implemented by other manufacturers without any coordination. Self-seeded codes should be contrasted to the notion of seeded codes in [3], which required a shared random seed as a common trusted setup between the encoder and decoder. Self-seeded codes are the strongest notion and directly imply the seemingly incomparable notions of randomized codes and seeded codes.

Minimal Assumptions

We show that any randomized code that goes beyond the Singleton bound with error detection tolerance $\rho > 1 - R$ or error correction tolerance $\rho > (1 - R)/2$ implies *one-way functions*, while any such seeded (and therefore also self-seeded) code implies *collision-resistant hash functions*.

Main Theorem

We construct randomized and self-seeded codes going beyond the Singleton bound, with essentially optimal parameters for both error detection and error correction, under the above respective minimal assumptions.

▶ **Theorem 1** (Informal). *Assuming one-way functions (resp. collision-resistant hash functions) there exist efficiently computable randomized (resp. self-seeded) error-detection and error-correction codes for arbitrary polynomial-time adversarial errors with the following parameters:*

- *Error Detection: For any constant $0 < \varepsilon < 1$, there is a code over a constant-sized alphabet with rate $R = 1 - \varepsilon$ that detects a $\rho = 1 - \varepsilon$ fraction of errors.*

[1] Randomized codes may also be secret-coin, where the adversary never sees the full randomness r. However, our construction will be public-coin and the randomness r can be easily recovered from the codeword c.

- *Error Correction: For any constants $0 < \varepsilon < 1$ and $0 \leq \rho < 1/2$, there is a code over a constant-sized alphabet with rate $R = 1 - \rho - \varepsilon$ that uniquely corrects from a ρ fraction of errors.*

If we assume sub-exponentially secure one-way functions (resp. collision-resistant hash functions), then the above properties hold even for sub-exponential time adversarial errors.

Recall that, in the case of error-detection, the Singleton bound says that the relative error tolerance $\rho \leq 1 - R$ necessarily degrades as the rate improves when considering worst-case errors; in contrast, for computationally bounded errors, our result simultaneously achieves the best possible error tolerance $\rho \approx 1$ and rate $R \approx 1$. In the case of error correction, the Singleton bound says the maximal error tolerance is $\rho \leq (1 - R)/2$ for worst-case errors, while for computationally bounded errors our result achieves twice as large error tolerance $\rho \approx 1 - R$, subject to the upper limit $\rho < 1/2$.

Our achieved parameters are essentially optimal, up to the arbitrarily small constant ε. This is clear for error detection, where we get the best possible rate $R \approx 1$ and error tolerance $\rho \approx 1$ simultaneously. For error-correction, one cannot tolerate $\rho \geq 1/2$ fraction of errors since an efficient adversarial channel can modify $1/2$ the codeword positions to a fresh encoding of a different message, causing decoding to fail with probability at least $1/2$.[2] Moreover, one cannot have rate $R > 1 - \rho$ since zeroing out the first ρ fraction of positions of the codeword would remove information about the message, causing decoding to fail.

We remark that, not only are one-way functions and collision-resistant hash functions the minimal assumptions needed to achieve the above results, they are considered some of the most standard and well-studied cryptographic assumptions/primitives. In particular, one-way functions are a minimal assumption necessary for essentially any non-trivial task in cryptography and are implied by all concrete cryptographic hardness assumptions. Collision-resistant hash functions are not known to be implied by one-way function, but can be constructed from essentially every concrete cryptographic hardness assumption, such as hardness of factoring, discrete logarithms, learning with errors, etc.

The closest related prior work of [3] achieves essentially the same parameters as our work, but only for the weaker notion of seeded codes, which require trusted setup, and only by assuming a strong form of "two-input correlation-intractable hash functions", which we do not know how to instantiate under any standard cryptographic hardness assumption. Our result for self-seeded codes yields a stronger primitive under significantly weaker minimal assumptions. Comparing the notions of randomized, seeded and self-seeded codes, it is clear that self-seeded codes are the strongest notion easily implying the other two, while randomized and seeded codes are seemingly incomparable. Interestingly, since we show that any non-trivial seeded codes imply collision-resistant hashing, which in turn implies self-seeded codes, we show that seeded and self-seeded codes are equivalent in terms of the needed assumptions.

Simultaneous Correction and Detection

While we stated separate results for error-detection and error-correction above, we can also simultaneously achieve a non-trivial combination of correction and detection. In particular, we can augment the previous theorem statement with the following:

[2] The works of [11, 12] overcome this limitation by having a stateful and secret keyed encoding procedure to ensure that an adversarial channel cannot generate/reuse valid codewords.

For any constants $0 < \varepsilon < 1$ and $0 \leq \rho_c < 1/2$, there is a code over a constant-sized alphabet with rate $R = 1 - \rho_c - \varepsilon$ that simultaneously uniquely corrects from a ρ_c fraction of errors and detects a $\rho_d = 1 - \rho_c - \varepsilon$ fraction of errors.

The above implies both of the previously stated results for error detection (by setting $\rho_c = 0$) and error correction (by ignoring ρ_d). However simultaneous error correction and detection is an interesting property, which is more than the sum of its parts. It gives a single decoding procedure that either outputs a message or $\perp$ (error detection symbol) and is guaranteed to uniquely recover the correct message if the error rate is $\leq \rho_c$ and to never output the wrong message as long as the error rate is $\leq \rho_d$. We can envision scenarios where correcting a small fraction of errors (e.g., $\rho_c = .1$) may suffice. Our result show that for such instances, we can have a code with a correspondingly large rate (e.g., $R \approx .9$) while simultaneously being able to detect a much larger fraction of errors (e.g., $\rho_d \approx .9$) at no additional cost.

Alphabet Size

We note that our results hold for large constant-size alphabets, but do not extend to the case of a binary alphabet. Giving optimal results for binary codes that detect/correct computationally bounded errors under minimal assumptions remains an interesting open problem. The results of [3] for seeded error correcting codes using a strong form of correlation intractable hash functions extend to the binary case and show that one can uniquely decode computationally bounded errors with essentially the same parameters as list-decoding for worst-case errors. It remains an open problem to achieve such results for randomized or self-seeded codes under any assumptions, and/or to to get provably secure seeded codes under standard cryptographic hardness assumptions.

1.2 Our Techniques

Our techniques are quite simple in retrospect. We start by showing how to construct self-seeded codes assuming collision-resistant hashing, and then discuss how to modify these results to get analogous ones for randomized codes from one-way functions. Throughout the introduction, we use "$\approx$" to denote that two values differ by an arbitrarily small constant.

Self-Seeded Error Detection with Sub-Exponential Alphabet

We start with a naive construction of self-seeded error-detection codes that requires a sub-exponentially large alphabet size, and then show how to reduce the alphabet size to some (large) constant. Let $\mathcal{H} = \{h_r \ : \ \{0,1\}^k \to \{0,1\}^\lambda\}_{r \in \{0,1\}^\lambda}$ be a collision-resistant hash function family with output length λ (security parameter), which we can think of as $\lambda = k^\alpha$ for some small constant $\alpha > 0$. Given a random public seed $r \leftarrow \{0,1\}^\lambda$, no polynomial-time adversary can find two messages $m \neq m'$ such that $h_r(m) = h_r(m')$.

The encoding algorithm $\mathrm{Enc}(m; r)$ takes a message $m \in \{0,1\}^k$ and parses it as $n = k/w$ symbols $m = (m_1, \ldots, m_n)$ over an alphabet $\Sigma_{\mathrm{data}} = \{0,1\}^w$ for some large $w = O(\lambda)$. It then "folds in" the seed r and hash of the message $h_r(m)$ into each message symbol to yield the codeword

$$c = (\ [m_1, r, h_r(m)]\ ,\ [m_2, r, h_r(m)]\ ,\ \ldots\ ,\ [m_n, r, h_r(m)]\ ) \in \Sigma^n$$

interpreted as n symbols over the larger alphabet $\Sigma = \{0,1\}^{w+2\cdot\lambda}$. The decoding algorithm $\mathrm{Dec}(c')$ gets

$$c' = (\ [m'_1, r'_1, y'_1]\ ,\ [m'_2, r'_2, y'_2]\ ,\ \ldots\ ,\ [m'_n, r'_n, y'_n]\ ) \in \Sigma^n,$$

which gives a candidate message $m' = (m'_1, \ldots, m'_n)$. It checks that the second part of each codeword symbol has the same common seed value $r'_1 = \cdots = r'_n$ and that the third part of each symbol has the same common hash value $y'_i = h_{r'_i}(m')$; if so it outputs m' else $\perp$. The code achieves rate $R = w/(w + 2\lambda) \approx 1$ by choosing a sufficiently large $w = O(\lambda)$. Moreover, it achieves error-tolerance $\rho = 1 - 1/n$. Unless the adversary changes *every* symbol of the codeword c, or finds a collision, decoding is guaranteed to output the correct message m or $\perp$. To see this, assume at least one symbol of the codeword remains unchanged and decoding outputs $m' \neq m$ for some $m' \neq \perp$. Then it must be the case that $r'_i = r, y'_i = h_r(m') = h_r(m)$ for all i and therefore the this gives a collision $m \neq m'$ such that $h_r(m) = h_r(m')$.

Self-Seeded Error Detection with Constant Alphabet

We reduce the alphabet size to a constant by applying a standard error-correcting code $(\mathrm{Enc}_{\mathrm{ctrl}}, \mathrm{Dec}_{\mathrm{ctrl}})$ over a constant sized alphabet to the small "control" value $(r, h_r(m))$. In particular, for some constants $w > v$, we use a code with alphabet $\Sigma_{\mathrm{ctrl}} = \{0,1\}^v$ such that $\mathrm{Enc}_{\mathrm{ctrl}} : \{0,1\}^{2\lambda} \to \Sigma^n_{\mathrm{ctrl}}$ maps a 2λ-bit message to a codeword of length $n = k/w$. The corresponding rate of the code is very small $R_{\mathrm{ctrl}} = O(\lambda/k) = o(1)$, but we will want large distance $\delta_{\mathrm{ctrl}} \approx 1$. Such codes can be constructed by concatenating Reed-Solomon codes with a brute-force inner code over the alphabet Σ_{ctrl} (see Theorem 13). We do not need efficient error-decoding for now, but only require an efficient way to recognize and valid codewords and invert them without errors. The encoding algorithm of our self-seeded code $\mathrm{Enc}(m; r)$ first computes $c_{\mathrm{ctrl}} = (c_{\mathrm{ctrl},1}, \ldots, c_{\mathrm{ctrl},n}) = \mathrm{Enc}_{\mathrm{ctrl}}((r, h_r(m))) \in \Sigma^n_{\mathrm{ctrl}}$. It interprets $m \in \{0,1\}^k$ as $m = (m_1, \ldots, m_n) \in \Sigma^n_{\mathrm{data}}$ consisting of $n = k/w$ symbols over the alphabet $\Sigma_{\mathrm{data}} = \{0,1\}^w$ and then "folds" the two vectors together to yield the final codeword

$$c = (\ [m_1, c_{\mathrm{ctrl},1}]\ ,\ \ldots\ ,\ [m_n, c_{\mathrm{ctrl},n}]\ ) \in \Sigma^n,$$

interpreted as n symbols in over the alphabet $\Sigma = \Sigma_{\mathrm{data}} \times \Sigma_{\mathrm{ctrl}} = \{0,1\}^{v+w}$. The decoding algorithm $\mathrm{Dec}(c')$ gets

$$c' = (\ [m'_1, c'_{\mathrm{ctrl},1}]\ ,\ \ldots\ ,\ [m'_n, c'_{\mathrm{ctrl},n}]\ ) \in \Sigma^n,$$

and recovers a candidate message $m' = (m'_1, \ldots, m'_n)$ as well as a candidate $c'_{\mathrm{ctrl}} = (c'_{\mathrm{ctrl},1}, \ldots, c'_{\mathrm{ctrl},n})$. It checks that c'_{ctrl} is a valid codeword and if so recovers (r', y') such that $c'_{\mathrm{ctrl}} = \mathrm{Enc}_{\mathrm{ctrl}}(r', y')$. If c'_{ctrl} is not a valid codeword or $h_{r'}(m') \neq y'$, it outputs $\perp$ else it outputs m'. Note that this generalizes the previous naive idea with a sub-exponentially large alphabet, which corresponds to using a repetition code for the control code. The rate of the overall code is $R = w/(w + v)$ which can be made arbitrarily close to 1 by choosing a sufficiently large constant $w > v$. The error-tolerance is $\rho = \delta_{\mathrm{ctrl}} \approx 1$. In particular, if c' is within distance ρ of c and c'_{ctrl} is a valid codeword then it must be the case that $c'_{\mathrm{ctrl}} = c_{\mathrm{ctrl}}$ and hence $r' = r, y' = h_r(m)$. In this case, the only way that that decoding can output some $m' \neq m$ is if the values form a hash collision $h_r(m') = h_r(m)$.

Self-Seeded Error Correction

To achieve error-correction, we use essentially the same construction as before, but additionally encode the data m using an efficiently list-decodable error-correcting code $(\mathrm{Enc}_{\mathrm{data}}, \mathrm{Dec}_{\mathrm{data}})$ over a constant size alphabet $\Sigma_{\mathrm{data}} = \{0,1\}^w$ for some large constant w. For any constant $\rho < 1/2$ we want this code to have rate $R_{\mathrm{data}} \approx 1 - \rho$ and to be list-decodable from a ρ fraction of errors. This can be accomplished using the codes of [5]. Moreover, we will now want the code $(\mathrm{Enc}_{\mathrm{ctrl}}, \mathrm{Dec}_{\mathrm{ctrl}})$ to efficiently uniquely decode from ρ errors, which

can be achieved with the same parameters as previously. Our self-seeded code $\mathrm{Enc}(m;r)$ computes $c_{\mathrm{ctrl}} = (c_{\mathrm{ctrl},1},\ldots,c_{\mathrm{ctrl},n}) = \mathrm{Enc}_{\mathrm{ctrl}}((r,h_r(m))) \in \Sigma_{\mathrm{ctrl}}^n$ as before, and also $c_{\mathrm{data}} = (c_{\mathrm{data},1},\ldots,c_{\mathrm{data},n}) = \mathrm{Enc}_{\mathrm{data}}(m) \in \Sigma_{\mathrm{data}}^n$. It then "folds" the two codewords together to yield the final codeword

$$c = (\ [c_{\mathrm{data},1}, c_{\mathrm{ctrl},1}]\ ,\ \ldots\ ,\ [c_{\mathrm{data},n}, c_{\mathrm{ctrl},n}]\ ) \in \Sigma^n$$

where $\Sigma = \Sigma_{\mathrm{data}} \times \Sigma_{\mathrm{ctrl}} = \{0,1\}^{v+w}$. The decoding algorithm $\mathrm{Dec}(c')$ gets

$$c' = (\ [c'_{\mathrm{data},1}, c'_{\mathrm{ctrl},1}]\ ,\ \ldots\ ,\ [c'_{\mathrm{data},n}, c'_{\mathrm{ctrl},n}]\ ) \in \Sigma^n.$$

It applies list-decoding on the data part c'_{data} of the received value c to recover a polynomial list of candidate messages $\{m^{(i)}\} = \mathrm{Dec}_{\mathrm{data}}(c'_{\mathrm{data}})$. It also uniquely decodes $(r',y') = \mathrm{Dec}_{\mathrm{ctrl}}(c'_{\mathrm{ctrl}})$. Then it outputs the first value $m^{(i)}$ such that $h_{r'}(m^{(i)}) = y'$, or outputs $\perp$ if none is found or if either of the decoding procedures fail. The rate of the overall code is $R = R_{\mathrm{data}} \cdot (w+v)/w \approx 1-\rho$ by choosing sufficiently large constants $w > v$. To analyze the error-correction, first observe that if the received value c' within relative distance ρ of $c = \mathrm{Enc}(m;r)$ then, by the correctness of list-decoding of the "data code", the correct message m will be in the decoded list with $m = m^{(i)}$ for some i, and by the correctness of unique decoding of the "control code" we have $(r',y') = (r,h_r(m))$. So the only way that error correction can fail is if there is another $m^{(j)} \neq m$ in the list such that $h_r(m^{(j)}) = h_r(m)$, but this gives a collision on the hash function.

Randomized Codes from One-Way Functions

Our constructions of *randomized* codes with error-detection and error-correction are identical to the above constructions of *self-seeded* codes, but instead of requiring the hash family h_r to be collision-resistant, we only need it to be a *universal one-way hash function (UOWHF).* A UOWHF ensures that a polynomial time adversary that selectively chooses a message m first and then gets the seed r cannot find a collision $m' \neq m$ such that $h_r(m) = h_r(m')$ except with negligible probability. Such UOWHFs can be constructed from one-way functions [13, 16]. Recall that the only difference between self-seeded and randomized codes is whether the encoded message m can be chosen adaptivley depending on the randomness r or selectively before r is chosen, which exactly matches the difference between collision-resistant hash functions and UOWHFs.

Necessity of Assumptions

We show that self-seeded codes beating the Singleton bound, with error-detection tolerance $\rho > 1-R$ or error-correction tolerance $\rho > (1-R)/2$, imply collision-resistant hash functions (CRHFs). Start by considering a self-seeded error-detection code $(\mathrm{Enc},\mathrm{Dec})$ with error-detection tolerance ρ. We claim that for every fixed subset S of $1-\rho$ fraction of codeword positions, the function $h_r(m) = \mathrm{Enc}(m;r)[S]$ that outputs the codeword symbols of $\mathrm{Enc}(m;r)$ in the positions indexed by S is collision-resistant. In particular, if there is an efficient adversary that, given r, can find a collision $m \neq m'$ such that $h_r(m) = h_r(m')$, it means that $\mathrm{Enc}(m;r)$ and $\mathrm{Enc}(m';r)$ are at distance $< \rho$ and therefore such an adversary can break error detection by choosing the message m and modifying $\mathrm{Enc}(m;r)$ to $\mathrm{Enc}(m';r)$. Moreover, if the rate of the code exceeds the Singleton bound with $R > 1-\rho$, then the functions h_r are compressing, making them CHRFs. To get the analogous result for error-correction, we simply note that ρ-error-correction implies (2ρ)-error-detection. In particular, if an efficient adversary given r can find two messages $m_0 \neq m_1$ such that $\mathrm{Enc}(m_0;r)$ and

$\mathrm{Enc}(m_1; r)$ are within relative distance 2ρ then it can find a "midpoint" c' which is at relative distance ρ from both $\mathrm{Enc}(m_0; r)$ and $\mathrm{Enc}(m_1; r)$. For one of $b \in \{0, 1\}$, modifying $\mathrm{Enc}(m_b; r)$ to c' necessarily causes decoding to output the incorrect message m_{1-b}.

We also show that randomized error-detection and error-correction codes beating the Singleton bound imply one-way functions. The argument is somewhat different from above since we don't want to assume that $\mathrm{Enc}(m; r)$ reveals r in the full. Instead, we rely on the fact that, if one-way functions don't exist, then we can efficiently sample "almost uniform" inverses of any efficient function [9]. Consider a random codeword $c = \mathrm{Enc}(m; r)$ for a random message m and randomness r. For any subset S of $1 - \rho$ fraction of codeword positions we can efficiently find an almost uniform inverse (m', r') of the function $f(m, r) = \mathrm{Enc}(m; r)[S]$ that outputs the codeword symbols in positions S. This means that $c = \mathrm{Enc}(m; r)$ and $c' = \mathrm{Enc}(m'; r')$ agree on the positions in S and therefore have relative distance $\leq \rho$. If the rate of the code exceeds the Singleton bound with $R > 1 - \rho$, then there is a non-negligible probability that $m \neq m'$ since f loses information about m. Therefore the channel that changes c to c' is efficient and defeats error detection by introducing ρ-fraction errors. Similarly the channel that changes a random subset of $\rho/2$ positions outside of S from c to c' defeats error correction since decoding is almost as likely to output m' as m.

Organization of the Paper

In Section 2 we give general notations and state the ingredients that we will use. In Section 3 we give formal definitions for randomized codes and self-seeded codes. In Section 4 we state and prove our main theorems. In Section 5 we show that randomized codes imply one-way functions. Due to space limitations, this extended abstract does not contain the proof that self-seeded codes imply collision resistance hash functions. The reader is referred to [22] for a full version that contains this proof.

2 Preliminaries and Ingredients

2.1 Notations

All logarithms are taken in base 2. Let poly stand for the set of all polynomials. Let PPT stand for non-uniform probabilistic polynomial-time. We note that while we rely on the non-uniform model of computation by default, all our results also hold if we were to define PPT in the uniform model. We use sans-serif (e.g., A) for PPT algorithms. A function $\nu \colon \mathbb{N} \mapsto [0, 1]$ is *negligible*, denoted $\nu(k) = \mathrm{neg}(k)$, if $\nu(k) < 1/p(k)$ for every $p \in \mathrm{poly}$ and large enough k. For a set S, let $x \leftarrow S$ denote that x was drawn uniformly from S. Given a distribution P, we use $x \leftarrow P$ to denote that x is chosen according to P.

Hamming distance

The Hamming distance between $x, y \in [q]^n$ is $\Delta(x, y) = |\{i : x_i \neq y_i\}|$. The relative Hamming distance between $x, y \in [q]^n$ is $\delta(x, y) = \frac{\Delta(x,y)}{n}$.

Distributions and random variables

The statistical distance (also known as, variation distance) of two distributions P and Q over a discrete domain X is defined by $\mathsf{SD}(P, Q) = \max_{S \subseteq X} |P(S) - Q(S)|$.

▶ **Definition 2** (Min-entropy). *A random variable X has min-entropy k, denoted $H_\infty(X) \geq k$, if $\max_x \mathbf{Pr}[X = x] \leq 2^{-k}$.*

We will also use the following lemma about average min-entropy:

▶ **Definition 3** (Average min-entropy). *Let (X, Y) be a pair of random variables. The average min-entropy of X conditioned on Y is:*

$$\tilde{H}_\infty(X|Y) = -\log\left[\mathop{\mathbf{E}}_{y\leftarrow Y}\left[2^{-H_\infty(X|Y=y)}\right]\right]$$

Average min-entropy gives a bound on the probability that an adversary that gets the value of Y will guess the value of X in a single attempt. We will use the following useful lemma that connects average min-entropy and min-entropy.

▶ **Lemma 4** ([2, 15]). *$\tilde{H}_\infty(X|Y) \geq H_\infty(X, Y) - v$, where 2^v is the number of elements in support of Y.*

2.2 One-Way Functions

▶ **Definition 5** (One-way functions). *A* PPT *algorithm* $\mathsf{F}\colon \{0,1\}^* \to \{0,1\}^*$ *is a* **one-way function** *if for every* PPT A *and every sufficiently large $k \in \mathbb{N}$,*

$$\Pr_{x\leftarrow\{0,1\}^k}\left[\mathsf{A}(1^k, \mathsf{F}(x)) \in \mathsf{F}^{-1}(\mathsf{F}(x))\right] \leq \mathrm{neg}(k).$$

Distributional one-way functions

We will also use the classical result of Impagliazzo and Luby on distributional one-way functions implying one-way functions.

▶ **Definition 6** (Distributional One-way functions). *A* PPT *algorithm* $\mathsf{F}\colon \{0,1\}^* \to \{0,1\}^*$ *is a* **distributional one-way function** *if there exists a constant $c > 0$ such that for every* PPT A *and every sufficiently large $k \in \mathbb{N}$,*

$$SD\Big(\big(\mathsf{A}(1^k, \mathsf{F}(X)), \mathsf{F}(X)\big), \big(X, \mathsf{F}(X)\big)\Big) \geq 1/k^c.$$

where X is sampled uniformly from $\{0,1\}^k$.

▶ **Theorem 7** (Distributional one-way functions imply one-way functions, [9]). *If there exists a distributional one-way function then there exists a one-way function.*

2.3 Universal One-Way Hash Functions

▶ **Definition 8** (Universal one-way hash functions). *For a security parameter $k \in \mathbb{N}$. A family of functions $\mathcal{H} = \{h_s\colon \{0,1\}^{n(k)} \to \{0,1\}^{\ell(k)}\}_{s\in\{0,1\}^{\ell(k)}}$ is a family of* **universal one-way hash functions** *(UOWHFs) if it satisfies:*

1. *Efficiency: Given $s \in \{0,1\}^{\ell(k)}$ and $x \in \{0,1\}^{n(k)}$, $h_s(x)$ can be evaluated in time* $\mathrm{poly}(n(k), \ell(k))$.
2. *Shrinking: $\ell(k) < n(k)$.*
3. *Target Collision Resistance: For every* PPT A *and sufficiently large $k \in \mathbb{N}$, if we consider the following randomized experiment:*
   - $(x, \mathrm{state}) \leftarrow \mathsf{A}(1^k) \in \{0,1\}^{n(k)} \times \{0,1\}^*$,
   - *Sampling* $s \leftarrow \{0,1\}^{\ell(k)}$,
   - $x' \leftarrow \mathsf{A}(\mathrm{state}, s)$

   Then, $\mathbf{Pr}\left[x \neq x' \wedge h_s(x) = h_s(x')\right] \leq \mathrm{neg}(k)$.

The notion of universal one-way hash functions (UOWHF) was first introduced by Naor and Yung [13]. Rompel [16] gave the first UOWHF construction from arbitrary one-way functions.

▶ **Theorem 9** (One-way functions imply universal one-way hash function, [13, 16]). *There is a black-box construction of universal one-way hash functions (UOWHFs) from one-way functions (OWFs). In particular, assuming OWFs, for any arbitrarily large constant c and arbitrarily small constant $\delta > 0$ there is a UOWHF with input length $n(k) = k^c$ and output/seed length $\ell(k) = k^\delta$.*

2.4 Collision Resistance Hashing

▶ **Definition 10** (Collision resistance hash functions (CRHF)). *For a security parameter $k \in \mathbb{N}$. A family of functions $\mathcal{H} = \{h_s \colon \{0,1\}^{n(k)} \to \{0,1\}^{\ell(k)}\}_{s \in \{0,1\}^{\ell(k)}}$ is a family of* **collision resistance hash functions (CRHFs)** *if it satisfies:*

1. *Efficiency: Given $s \in \{0,1\}^{\ell(k)}$ and $x \in \{0,1\}^{n(k)}$, $h_s(x)$ can be evaluated in time $\mathrm{poly}(n(k), \ell(k))$.*
2. *Shrinking: $\ell(k) < n(k)$.*
3. *Collision Resistance: For every* PPT A *and sufficiently large $k \in \mathbb{N}$,*

$$\Pr_{\substack{s \leftarrow \{0,1\}^k, \\ x, x' \leftarrow \mathsf{A}(s)}} [x \neq x' \wedge h_s(x) = h_s(x')] \leq \mathrm{neg}(k).$$

We mention that the parameters of CRHFs above do not matter much: any "non-trivial" CRHF implies an "arbitrarily good" CRHF. That is, start with any CRHF with arbitrarily large polynomial seed/output length $\ell(k)$ that has even 1-bit compression $n(k) = \ell(k) + 1$. Then, for any arbitrarily small constant $\delta > 0$ and arbitrarily large constant c, we can also get a CRHF with small seed/output length $\ell(k) = k^\delta$ and large input length $n(k) = k^c$. Therefore, when we assume the existence of CRHFs we without loss of generality assume the above "arbitrarily good" CRHF parameters.[3]

2.5 Standard Correcting/Detecting Codes

We now define error-correcting codes for information-theoretic adversaries also called Hamming channels. In this section, a code is a pair $(\mathrm{Enc}, \mathrm{Dec})$ of deterministic encoding and decoding algorithms, and different notions are obtained by considering the requirements of the decoding algorithm.

▶ **Definition 11** (Standard codes for Hamming channels). *Let k, n, q be parameters and let $\mathrm{Enc} : \{0,1\}^k \to [q]^n$ be a function. We say that* Enc *is an encoding function for a code that:*

- **decodes** *d* **errors**, *if there exists a function $\mathrm{Dec} : [q]^n \to \{0,1\}^k$ such that for every $m \in \{0,1\}^k$ and every $v \in [q]^n$ with $\Delta(\mathrm{Enc}(m), v) \leq d$, we have $\mathrm{Dec}(v) = m$.*
- *L*-**list-decodes** *d* **errors**, *if the function* Dec *is allowed to output a list of size at most L, and for every $m \in \{0,1\}^k$ and every $v \in [q]^n$ with $\Delta(\mathrm{Enc}(m), v) \leq d$, we have $m \in \mathrm{Dec}(v)$.*
- **detects** *d* **errors**, *if there exists a function $\mathrm{Dec} : [q]^n \to \{0,1\}^k \cup \{\perp\}$ such that for every $m \in \{0,1\}^k$, $\mathrm{Dec}(\mathrm{Enc}(m)) = m$ and for every $v \in [q]^n$ such that $v \neq \mathrm{Enc}(m)$ and $\Delta(\mathrm{Enc}(m), v) \leq d$, we have $\mathrm{Dec}(v) = \perp$.*

[3] We can decrease the seed/output length by changing the security parameter from k to k^ε for a sufficiently small ε and noting that poly/negl in k^ε is the same as poly/negl in k. We can increase the input length arbitrarily using the Merkle-Damgard transform.

The rate of the code is the ratio of the message length and output length of Enc*, where both lengths are measured in bits. That is the rate* $R = \frac{k}{n \cdot \log q}$.

*A code (*Enc, Dec*), is* **explicit** *if* Enc *and* Dec *run in polynomial time. (Naturally, this makes sense only for a family of encoding and decoding/detecting functions with varying message length* k *block length* $n(k)$*, and alphabet sizes* $q(k)$*).*[4]

We will also use the following construction by Guruswami and Rudra [5] for optimal list-decodable codes over a constant alphabet.

▶ **Theorem 12** ([5])**.** *For every* $0 < R < 1$*, every sufficiently small* $\varepsilon > 0$*, there is an explicit family of codes over an alphabet of size* $2^{O(\varepsilon^{-4} \log(\frac{1}{\varepsilon}))}$ *that have rate at least* R *and which can be list decoded up to a fraction* $1 - R - \varepsilon$ *of errors in polynomial time, with a list of size* $n^{O(\varepsilon^{-1} \log(\frac{1}{\varepsilon}))}$.[5]

It is folklore that for every $\varepsilon > 0$ by using code concatenation it is possible to construct standard codes with distance $d \geq 1 - \varepsilon$ and rate $R = o(1)$ for any sufficiently large k with an alphabet size that depends on ε. We make this formal in the following Theorem (see for example the book by Guruswami, Rudra, and Sudan [6] for more details).

▶ **Theorem 13.** *For every sufficiently small* $0 < \varepsilon$*, and every computable function* $n : \mathbb{N} \mapsto \mathbb{N}$ *such that for every sufficiently large* k*,* $n(k) > k^c$*, (for a universal constant* $c > 1$*) there exists an explicit standard code* $(\mathrm{Enc}, \mathrm{Dec})$*, for* $\mathrm{Enc}\colon \{0,1\}^k \to [q]^{n(k)}$ *and* $\mathrm{Dec}\colon [q]^{n(k)} \to \{0,1\}^k$ *with alphabet* $q = 2^{(1/\varepsilon^3)}$ *such that the code has distance* $d(k) > n(k) - \varepsilon \cdot n(k)$ *and (uniquely) decodes from* $d(k)/2 - 1$ *errors.*

3 Definitions for Randomized Codes and Self-Seeded Codes

In this section, we formally define randomized codes and self-seeded codes, and their related notions of error detection and correction.

3.1 Randomized Codes

We start by giving a formal definition of randomized encoding and decoding algorithms. This syntax will be also shared with our notion of self-seeded codes defined in the next section.

▶ **Definition 14** (Randomized codes)**.** *Let* $n, q : \mathbb{N} \mapsto \mathbb{N}$ *be poly-time computable functions. An* (n, q)**-randomized** *code is a pair* $(\mathrm{Enc}, \mathrm{Dec})$ *such that:*

- Enc *is a* PPT *algorithm that on input* $m \in \{0,1\}^k$ *outputs a string in* $c \in [q(k)]^{n(k)}$*. We use* $\mathrm{Enc}_r(m)$ *to denote the instantiation of* $\mathrm{Enc}(m)$ *when using the string* r *as random coins.*
- Dec *is a* PPT *algorithm that on input* $z \in [q(k)]^{n(k)}$ *outputs a string* $\widehat{m} \in \{0,1\}^k \cup \{\perp\}$.

[4] We point out that traditionally codes are indexed by the codeword length (that is, k and q are chosen as a function of the codeword size n). However, for the sake of consistency with the rest of our definitions, we chose to define the codeword length n and alphabet q as a function of the input size k.

[5] We remark that while Theorem 12 does not explicitly mention that the rate can be achieved for every sufficiently large message size k, and instead just claims that it holds for infinitely often choices of k. A careful examination of the construction and proof by [5] reveals that the argument and proof hold for every sufficiently large k, given a suitable expander graph. An explicit construction for such graphs were given for every sufficiently large k in [10] (see Lemma 2.7 for a formal proof). Thus, we assume that Theorem 12 holds for every sufficiently large k.

For a fixed $k \in \mathbb{N}$, the rate of the code for k is defined to be $R_k = \frac{k}{\log(q)\cdot n(k)}$. When $R = \lim_{k\to\infty} R_k \in [0,1]$ is well defined we say that R is the rate of the code. We omit all or a subset of the functions n and q when they are clear from the context.

We now define a notion of reliable codes for randomized codes, which essentially states that it is possible to recover the message given a codeword (that was not damaged by error).

▶ **Definition 15** (Codes with reliable decoding). *A randomized code* $(\mathrm{Enc}, \mathrm{Dec})$ *is said to be* **perfectly reliable** *if for every $k \in \mathbb{N}$ and $m \in \{0,1\}^k$,* $\mathbf{Pr}\left[\mathrm{Dec}(\mathrm{Enc}(m)) = m\right] = 1$.

Intuitively, in randomized codes, each time the sender wants to send a message, it samples fresh randomness on the fly and uses it to encode the message. We make this formal in the definition below.

▶ **Definition 16** (Error detection/correction for randomized codes). *Let $n, \ell, d : \mathbb{N} \mapsto \mathbb{N}$ be poly-time computable functions and let* $(\mathrm{Enc}, \mathrm{Dec})$ *be a randomized code. For a* PPT *algorithm* A *and $k \in \mathbb{N}$, consider the following randomized experiment:*

1. $(m, \mathrm{state}) \leftarrow \mathsf{A}(1^k) \in \{0,1\}^k \times \{0,1\}^*$
2. *Compute* $c \leftarrow \mathrm{Enc}(m)$ *(by sampling* $r \leftarrow \{0,1\}^{\ell(k)}$ *and computing* $\mathrm{Enc}_r(m)$*).*
3. $z \leftarrow \mathsf{A}(m, \mathrm{state}, c)$

We say that the code:

- *is* **reliable** *if for every* PPT *algorithm* A *and every sufficiently large $k \in \mathbb{N}$:* $\mathbf{Pr}\left[\mathrm{Dec}(c) \neq m\right] \leq \mathrm{neg}(k)$.[6]
- **detects** *d-errors if it is a* **reliable** *code, and for every* PPT *algorithm* A *and every sufficiently large $k \in \mathbb{N}$:* $\mathbf{Pr}\left[\Delta(c,z) \leq d(k) \wedge \mathrm{Dec}(z) \notin \{\perp, m\}\right] \leq \mathrm{neg}(k)$.
- **corrects** *d-errors if for every* PPT *algorithm* A *and every sufficiently large $k \in \mathbb{N}$:* $\mathbf{Pr}\left[\Delta(c,z) \leq d(k) \wedge \mathrm{Dec}(z) \neq m\right] \leq \mathrm{neg}(k)$.

For a constant $\rho \in [0,1]$, the code detects or corrects ρ-relative errors, if for every sufficiently large $k \in \mathbb{N}$, $\rho \leq d(k)/n(k)$.

Note that by definition if a code corrects d-errors for $d > 0$ then the code is reliable. In addition, our definition of error detection permits also error correction (that is, when errors are induced, the decoding algorithm can either return $\perp$ or successfully decode and return the original message m). By doing so we can achieve simultaneous error detection and correction (that is, the same decoding function can at the same time satisfy the notion of detection and correction defined above).

3.2 Self-Seeded Codes

We now formally, define the notion of self-seeded codes. A self-seeded code is a randomized code that shares the same syntax as defined in 14. However, a self-seeded code provides a strictly stronger notion of security as compared to the notion of error correction and detection for randomized codes stated above in Definition 16.

▶ **Definition 17** (Error detection/correction for self-seeded codes). *Let $n, \ell, d : \mathbb{N} \mapsto \mathbb{N}$ be poly-time computable functions. A randomized code is said to be a* **self-seeded** *code if the following holds: For a* PPT *algorithm* A *and $k \in \mathbb{N}$, consider the following randomized experiment:*

[6] This is a weaker variant of Definition 15. Our construction achieves the stronger variant of perfect reliability. However, our lower bounds results are stronger since they apply to this weaker notion.

1. $r \leftarrow \{0,1\}^{\ell(k)}$
2. $(m, z) \leftarrow \mathsf{A}(r)$

We say that the code:

- *is* **reliable** *if for every* PPT *algorithm* A *and every sufficiently large* $k \in \mathbb{N}$*:* $\mathbf{Pr}\left[\mathrm{Dec}(\mathrm{Enc}_r(m)) \neq m\right] \leq \mathrm{neg}(k)$.
- **detects** *d-errors if it is a* **reliable** *code, and for every* PPT *algorithm* A *and every sufficiently large* $k \in \mathbb{N}$*:* $\mathbf{Pr}\left[\Delta(\mathrm{Enc}_r(m), z) \leq d(k) \wedge \mathrm{Dec}(z) \notin \{\perp, m\}\right] \leq \mathrm{neg}(k)$.
- **corrects** *d-errors if for every* PPT *algorithm* A *and every sufficiently large* $k \in \mathbb{N}$*:* $\mathbf{Pr}\left[\Delta(\mathrm{Enc}_r(m), z) \leq d(k) \wedge \mathrm{Dec}(z) \neq m\right] \leq \mathrm{neg}(k)$.

For a constant $\rho \in [0,1]$*, the code detects or corrects* ρ*-relative errors, if for every sufficiently large* $k \in \mathbb{N}$*,* $\rho \leq d(k)/n(k)$.

Self-seeded code enables the sender to sample its randomness once, fix it, and then use this same randomness for encoding multiple messages (this contrasts with the standard notion of randomized codes, which requires the sender to generate fresh randomness for each message in real time). We stress that after being fixed, this randomness is known to any potential adversary but is *not* given to the decoding algorithm.[7] Still, we require that no efficient adversary can find a message and a close error vector that leads to a decoding error.

Indeed, self-seeded codes offer a stronger adaptive notion of security, where the adversary can choose the message m adaptively based on the randomness fixed by the sender. In contrast, standard randomized codes require the sender to choose fresh randomness after the adversary has decided on the message.

4 Optimal Randomized Codes and Self-Seeded Codes

We now state our main theorems for randomized codes and self-seeded codes. Our results show that it is possible to construct codes that bypass the Singleton bound for computationally bounded channels under minimal cryptographic assumptions. Specifically, for the randomized codes construction, we assume the existence of one-way functions, and for the self-seeded codes construction, we assume the existence of a collision resistance hash function.

Main theorem

We state our result for randomized codes and self-seeded codes together in the following theorem.

▶ **Theorem 18** (Randomized codes [resp., self-seeded codes] with constant alphabet size)**.** *Assuming the existence of one-way functions [resp., collision resistance hash functions], the following holds: For every constant* $0 \leq p < 1/2$ *and every sufficiently small constant* $0 < \varepsilon$*, there exists a* (q, n)*-randomized [resp.,* (q, n)*-self-seeded code] code with rate* $R = 1 - p - \varepsilon$ *that simultaneously (uniquely) corrects* p *relative errors and detects from* $1 - p - \varepsilon$ *relative errors. Moreover, the alphabet size* q *is a constant that depends on* ε*.*

The proof of Theorem 18 appears in Section 4.2 and relies on our construction described in Section 4.1.

[7] We note that our notion of self-seeded codes is strictly stronger than the notion of "seeded codes" defined in [3]. Seeded codes (unlike self-seeded codes), requires the encoding and decoding algorithms to share a random seed. Thus, unlike self-seeded codes, seeded codes necessitate a trusted setup between the encoding and decoding.

▶ Remark 19. If the underlying cryptographic primitives assumed in our theorem statements (that is, one-way functions for randomized codes and collision resistance hash functions for self-seeded codes) have sub-exponential security, then our constructed codes in Theorem 18 will also have sub-exponential failure probability for correction and detection against channels that are computable by sub-exponential size circuits. This follows since all our reductions run in polynomial time and have a polynomial loss in advantage. However, for the sake of simplicity, we focus on the case of polynomial/negligible security in our formal statements and proofs.

4.1 Construction

In this section, we describe the construction that we will use to prove Theorem 18.

We now describe the encoding and decoding algorithms. In Figure 1, we state the parameters and ingredients that are needed to define our encoding and decoding algorithms $(\mathrm{Enc}, \mathrm{Dec})$ that are specified in Figure 2. The encoding and decoding algorithms are essentially the same for randomized codes and for self-seeded codes. In both cases the encoding algorithm Enc takes as input a message m and some value r. For randomized codes, a fresh random r is chosen every time we send a new message, and for the self-seeded codes r is sampled once (and is given to any possible adversary), and this single fixed r could be used in the encoding of multiple messages.

In both cases, the decoding algorithm Dec only takes as input a possibly damaged codeword z. In particular, the decoding algorithm is not provided with the seed r.

4.2 Proof of Theorem 18

In this section, we prove Theorem 18. We make use of the following technical lemma. The proof for the lemma is given below in Section 4.2.1.

▶ **Lemma 20.** *If there are ingredients and parameters that satisfy the conditions stated in Figure 1. Moreover, if the hash family $\mathcal{H}$ specified in Figure 1 is a family of universal one-way hash functions [resp., collision resistance hash functions]. Then, the code* $(\mathrm{Enc}, \mathrm{Dec})$ *as specified in Figure 2 is a (q, n)-randomized code [resp., (q, n)-self-seeded code]. Moreover, the code is perfectly reliable, has alphabet size $q = q_{\mathrm{ctrl}} \cdot q_{\mathrm{data}}$, rate $R = R_{\mathrm{data}} \cdot \frac{\log(q_{\mathrm{data}})}{\log(q_{\mathrm{data}})+\log(q_{\mathrm{ctrl}})}$, and the decoding algorithm simultaneously,*

- *detects from $d_{\mathrm{ctrl}} - d_{\mathrm{data}}$ errors, and*
- *corrects from d_{data} errors.*

Proof of Theorem 18. Note that one-way functions imply universal one-way hash functions that satisfy the conditions stated for the hash family $\mathcal{H}$ in Figure 1 (see, Theorem 9).

Given a sufficiently small $0 < \varepsilon$, let $\hat{\varepsilon} = \varepsilon/4$ and apply Theorem 12 for $\hat{\varepsilon}$ and $R_{\mathrm{data}} = 1 - p - \hat{\varepsilon}$ to get a standard code $(\mathrm{Enc}_{\mathrm{data}}, \mathrm{Dec}_{\mathrm{data}})$ with rate $R_{\mathrm{data}} = 1 - p - \hat{\varepsilon}$, codeword length n_{data} alphabet size $q_{\mathrm{data}} = 2^{O(\hat{\varepsilon}^{-5})}$. Moreover, $(\mathrm{Enc}_{\mathrm{data}}, \mathrm{Dec}_{\mathrm{data}})$ list decodes from $d_{\mathrm{data}} = p \cdot n_{\mathrm{data}}$ errors (that is, from p relative errors) with a list size that depends only on $\hat{\varepsilon}$ and k. Note also that we can increase the alphabet size by bundling symbols together without damaging the list decoding and the rate R_{data} of the code, and so, we can assume that $\log(q_{\mathrm{data}})$ is of size at least $\hat{\varepsilon}^{-5}$. To summarize $(\mathrm{Enc}_{\mathrm{data}}, \mathrm{Dec}_{\mathrm{data}})$ meets the conditions stated in Figure 1 for "data code" for the parameters R_{data} and d_{data} and $\log(q_{\mathrm{data}}) > \hat{\varepsilon}^{-5}$.

Apply theorem 13 for $\hat{\varepsilon}$ to get a code $(\mathrm{Enc}_{\mathrm{ctrl}}, \mathrm{Dec}_{\mathrm{ctrl}})$ with codeword length $n_{\mathrm{data}} = n_{\mathrm{ctrl}}$, alphabet size at most $q_{\mathrm{ctrl}} = 2^{\hat{\varepsilon}^{-3}}$ that has distance $d_{\mathrm{ctrl}} > n_{\mathrm{ctrl}}(1 - \hat{\varepsilon})$ and efficient decoding of up to $d_{\mathrm{ctrl}}/2 - 1$ errors. Note that for a sufficiently small ε (and $\hat{\varepsilon}$) it holds that $d_{\mathrm{ctrl}} > 2d_{\mathrm{data}} + 1$. This means that $(\mathrm{Enc}_{\mathrm{ctrl}}, \mathrm{Dec}_{\mathrm{ctrl}})$ meets the conditions stated in Figure 1 for the "control code" with parameters d_{ctrl} and $\log(q_{\mathrm{ctrl}}) < \hat{\varepsilon}^{-3}$.

List of ingredients and parameters: Let $q, q_{\text{data}}, q_{\text{ctrl}}, d_{\text{data}}, d_{\text{ctrl}}, \ell, n : \mathbb{N} \mapsto \mathbb{N}$ and $R_{\text{data}} : \mathbb{N} \mapsto \mathbb{R}$ be poly-time computable functions.

Data code: A family of standard codes $(\text{Enc}_{\text{data}}, \text{Dec}_{\text{data}})$ such that $\text{Enc}_{\text{data}} \colon \{0,1\}^k \to [q_{\text{data}}]^n$ list-decodes from d_{data} errors with rate R_{data} and polynomial list size.

A hash family: $\mathcal{H} = \{h_s \colon \{0,1\}^k \to \{0,1\}^\ell\}_{s \in \{0,1\}^\ell}$ that is efficiently computable. Where we require that $2\ell := \ell' \leq k^{c_\ell}$ (where $c_\ell < 1$ is a fixed universal constant).

Control code: Code with large distance: A family of standard codes $(\text{Enc}_{\text{ctrl}}, \text{Dec}_{\text{ctrl}})$, such that $\text{Enc}_{\text{ctrl}} \colon \{0,1\}^{\ell'} \to [q_{\text{ctrl}}]^n$ has distance larger than $d_{\text{ctrl}} > 2 \cdot d_{\text{data}} + 1$ and efficiently uniquely decodes from $d_{\text{ctrl}}/2 - 1$ errors. (Note that we require the codeword size n to be the same in Enc_{data} and Enc_{ctrl}).

Figure 1 Parameters and Ingredients for folding construction.

Encoding algorithm

We define an encoding function $\text{Enc} : \{0,1\}^k \times \{0,1\}^\ell \to [q]^n$:
Input: A message $m \in \{0,1\}^k$. And coins $r \in \{0,1\}^\ell$.

Output: A codeword $c \in [q]^n$ for $q := q_{\text{ctrl}} \cdot q_{\text{data}}$.

Operation :

Encode the data: Compute $c_{\text{data}} = \text{Enc}_{\text{data}}(m) \in [q_{\text{data}}]^n$.
Hash the encoded data: Compute $y = h_r(m)$ and set $w := (y, r)$.
Encode the control: Compute $c_{\text{ctrl}} = \text{Enc}_{\text{ctrl}}(w) \in [q_{\text{ctrl}}]^n$
Fold the control and data codes: Compute $c \in ([q_{\text{data}}] \times [q_{\text{ctrl}}])^n$, where for every $i \in [n]$, $c_i = ((c_{\text{data}})_i, (c_{\text{ctrl}})_i)$. That is, c_i contains the i^{th} symbol of c_{data} and c_{ctrl}.
Output: c.

Decoding algorithm

We define a decoding function $\text{Dec} : [q]^n \to \{0,1\}^k \cup \{\perp\}$:
Input: A received word $z \in [q]^n$.

Output: A message $\widehat{m} \in \{0,1\}^k \cup \{\perp\}$.

Operation :

Unfold the control and data: For every $i \in [n]$ we view $z_i = ((z_{\text{data}})_i, (z_{\text{ctrl}})_i) \in [q_{\text{data}}] \times [q_{\text{ctrl}}]$ and compute $z_{\text{data}} = ((z_{\text{data}})_1, \cdots, (z_{\text{data}})_n)$ and $z_{\text{ctrl}} = ((z_{\text{ctrl}})_1, \cdots, (z_{\text{ctrl}})_n)$.
Uniquely decode the control part: $\widehat{w} = (\widehat{y}, \widehat{r}) = \text{Dec}_{\text{ctrl}}(z_{\text{ctrl}})$.
List decode the data part: Compute a list of message candidates $\text{List} = \text{Dec}_{\text{data}}(z_{\text{data}})$, by applying the list decoding for d_{data} errors.
Check control distance: If $\Delta(\text{Enc}_{\text{ctrl}}(\widehat{w}), z_{\text{ctrl}}) > d_{\text{data}}$ then output $\perp$.
Check consistency: Output the first message $\widehat{m} \in \text{List}$ for which $h_{\widehat{r}}(\widehat{m}) = \widehat{y}$. If no such message exists output $\perp$.

Figure 2 Encoding and decoding algorithm.

Thus, we can apply Lemma 20 for $(\mathrm{Enc}_{\mathrm{data}}, \mathrm{Dec}_{\mathrm{data}})$, $(\mathrm{Enc}_{\mathrm{ctrl}}, \mathrm{Dec}_{\mathrm{ctrl}})$ and $\mathcal{H}$ to get a (q, n)-randomized code when we take $\mathcal{H}$ to be a universal one-way hash family and (q, n)-self-seeded code if $\mathcal{H}$ is a collision resistance hash family. In both cases, it follows that

- $n = n_{\mathrm{data}}$
- $q = q_{\mathrm{data}} \cdot q_{\mathrm{ctrl}} = 2^{O(\varepsilon^{-5})}$,
- for every sufficiently small ε (and $\hat{\varepsilon}$), $R = R_{\mathrm{data}} \cdot \frac{\log(q_{\mathrm{data}})}{\log(q_{\mathrm{data}})+\log(q_{\mathrm{ctrl}})} > (1 - p - \hat{\varepsilon}) \cdot \frac{1}{1+\hat{\varepsilon}} > (1 - p - \varepsilon)$.
- The code detects from $d_{\mathrm{ctrl}} - d_{\mathrm{data}} = n(1 - \hat{\varepsilon}) - np > n(1 - p - \varepsilon)$ errors. That is, it detects from $1 - p - \varepsilon$ relative errors.
- The code corrects $d_{\mathrm{data}} = pn$ errors. That is, it corrects up to p relative errors.

This concludes the proof of Theorem 18. ◀

4.2.1 Proof of Lemma 20

Let $\mathrm{Enc}_{\mathrm{data}}, \mathrm{Enc}_{\mathrm{ctrl}}, h_r$, be as defined in Figure 1 and let $(\mathrm{Enc}, \mathrm{Dec})$ be as defined in Figure 2. By construction, $(\mathrm{Enc}, \mathrm{Dec})$ is perfectly reliable. This holds since for every fixed randomness r if no errors are applied, all ingredients in the construction are deterministic, and standard codes are by definition perfectly reliable. Similarly, by construction $q = q_{\mathrm{ctrl}} \cdot q_{\mathrm{data}}$ and

$$R = \frac{k}{n \cdot \log(q)} = \frac{R_{\mathrm{data}} \cdot (n \cdot \log(q_{\mathrm{data}}))}{n \cdot (\log(q_{\mathrm{ctrl}}) + \log(q_{\mathrm{data}}))} = R_{\mathrm{data}} \cdot \frac{\log(q_{\mathrm{data}})}{\log(q_{\mathrm{data}}) + \log(q_{\mathrm{ctrl}})}.$$

We now prove the rest of Lemma 20. For simplicity, we focus on the randomized codes case, the proof for the self-seeded codes is essentially the same and follows along similar lines.[8]

Consider the following randomized experiment for a PPT algorithm A and $k \in \mathbb{N}$:

1. $(m, \mathrm{state}) \leftarrow \mathsf{A}(1^k) \in \{0,1\}^k \times \{0,1\}^*$
2. Compute $c \leftarrow \mathrm{Enc}_r(m)$ (by sampling $r \leftarrow \{0,1\}^{\ell(k)}$).
3. $z \leftarrow \mathsf{A}(m, \mathrm{state}, c)$

We make use of the following notation: Let $c_{\mathrm{data}} = \mathrm{Enc}_{\mathrm{data}}(m)$, $w = (h_r(m), r)$ and $c_{\mathrm{ctrl}} = \mathrm{Enc}_{\mathrm{ctrl}}(w)$. Recall that $z \in ([q_{\mathrm{data}}] \times [q_{\mathrm{ctrl}}])^n$, and define $z_{\mathrm{data}} = ((z_{\mathrm{data}})_1, \cdots, (z_{\mathrm{data}})_n)$ and $z_{\mathrm{ctrl}} = ((z_{\mathrm{ctrl}})_1, \cdots, (z_{\mathrm{ctrl}})_n)$ such that for every $i \in [n]$, $z_i = ((z_{\mathrm{data}})_i, (z_{\mathrm{ctrl}})_i)$. Let $\widehat{w} = (\widehat{y}, \widehat{r}) = \mathrm{Dec}_{\mathrm{ctrl}}(z_{\mathrm{ctrl}})$ and $\mathrm{List} = \mathrm{Dec}_{\mathrm{data}}(z_{\mathrm{data}})$.

Error correction

To prove the error correction property, we need to show that

$$\mathbf{Pr}\left[\Delta(c, z) \leq d_{\mathrm{data}} \wedge \mathrm{Dec}(z) \neq m\right] \leq \mathrm{neg}(k).$$

We make use of the following claims:

▷ Claim 21. If $\Delta(c, z) \leq d_{\mathrm{data}}$ then $m \in \mathrm{List}$ and $w = \widehat{w}$.

Proof of Claim 21. If $\Delta(c, z) \leq d_{\mathrm{data}}$ it follows that $\Delta(c_{\mathrm{ctrl}}, z_{\mathrm{ctrl}}) \leq d_{\mathrm{data}}$ and $\Delta(c_{\mathrm{data}}, z_{\mathrm{data}}) \leq d_{\mathrm{data}}$. Since $(\mathrm{Enc}_{\mathrm{ctrl}}, \mathrm{Dec}_{\mathrm{ctrl}})$ is a standard code that uniquely decodes from $d_{\mathrm{ctrl}}/2 - 1 \geq d_{\mathrm{data}}$ errors it holds that $w = w'$. Similarly, since $(\mathrm{Enc}_{\mathrm{data}}, \mathrm{Dec}_{\mathrm{data}})$ is a standard code that list-decodes from d_{data} errors it follows that $m \in \mathrm{List}$. ◁

[8] The main difference is that we consider an adaptive adversary (that is, first $r \leftarrow \{0,1\}^{\ell(k)}$ and then $(m, z) \leftarrow \mathsf{A}(r)$) and we reduce to the security of the CRH instead of the security of the UOWHF, otherwise, the proof is identical.

▷ Claim 22. $\forall \widehat{m} \in \text{List}$, if $\widehat{m} \neq m$ then $\mathbf{Pr}\left[h_r(\widehat{m}) = y\right] \leq \text{neg}(k)$.

Proof of Claim 22. Follows directly from the security of the universal one-way hash function, as otherwise, we can use A to find collisions in h contradicting the collision resistance property. ◁

By Claim 21, $m \in \text{List}$ and $w = \widehat{w}$. Thus, by Claim 22, with all but negligible probability, m is the only message in List for which $h_r(m) = y$ (passing the "Check consistency" step in the decoding algorithm) which implies $\text{Dec}(z) = m$.

Error detection

To prove the error detection property, we need to show that

$$\mathbf{Pr}\left[\Delta(c, z) \leq d_{\text{ctrl}} - d_{\text{data}} \wedge \text{Dec}(z) \notin \{m, \perp\}\right] \leq \text{neg}(k).$$

We make use of the following Claim.

▷ Claim 23. If $\Delta(c, z) \leq d_{\text{ctrl}} - d_{\text{data}}$ and $\text{Dec}(z) \neq \perp$ then $w = \widehat{w}$.

Proof. Let $\widehat{c}_{\text{ctrl}} = \text{Enc}_{\text{ctrl}}(\widehat{w})$ and note that the by the "Check control distance" step in the decoding algorithm (see Figure 2), if $\text{Dec}(z) \neq \perp$ it follows that $\Delta(z_{\text{ctrl}}, \widehat{c}_{\text{ctrl}}) \leq d_{\text{data}}$. Moreover, by construction, if $\Delta(c, z) \leq d_{\text{ctrl}} - d_{\text{data}}$ it follows that $\Delta(c_{\text{ctrl}}, z_{\text{ctrl}}) \leq d_{\text{ctrl}} - d_{\text{data}}$. Thus, if $\text{Dec}(z) \neq \perp$ and $\Delta(c, z) \leq d_{\text{ctrl}} - d_{\text{data}}$ by the triangle inequality

$$\Delta(c_{\text{ctrl}}, \widehat{c}_{\text{ctrl}}) \leq d_{\text{ctrl}}.$$

Since c_{ctrl} and $\widehat{c}_{\text{ctrl}}$ are both valid codewords of the standard code $(\text{Enc}_{\text{ctrl}}, \text{Dec}_{\text{ctrl}})$ that has distance d_{ctrl}, it follows that $w = \widehat{w}$. ◁

By Claim 23 if $\text{Dec}(z) \neq \perp$ then $w = \widehat{w}$. This concludes the proof, since by Claim 22, if $w = w'$, with all but negligible probability, m is the only valid message that can be outputted by $\text{Dec}(z)$.

5 Randomized Codes Imply One Way Functions

We now show that non-trivial randomized codes imply the existence of one-way functions. For large alphabets, this is indeed optimal since as we have seen in our construction it is possible to construct optimal randomized codes assuming only universal one-way functions (which can be constructed from one-way functions).

The following theorem states that if a randomized code has a rate that bypasses the singleton bound, it implies one-way functions.

▶ **Theorem 24** (Randomized codes with rate beating the Singleton bound imply one way functions). *Let* (Enc, Dec) *be a randomized code with rate* $0 \leq R < 1$. *If one of the following holds:*

- *The code detects* ρ_0*-relative errors and* $\rho_0 > 1 - R + \frac{1}{n}$.
- *The code corrects from* ρ_1*-relative errors and* $2\rho_1 > 1 - R + \frac{2}{n}$.

Then there exists one-way functions.

The proof will make use of the following lemma.

▶ **Lemma 25** (Non-trivial error detection/correction for randomized codes implies one-way functions). *Let* (Enc, Dec) *be an* (n, q)*-randomized code that has at least one of the following properties:*

- *The code detects d-errors.*
- *The code corrects $\lceil d/2 \rceil$-errors.*

Let $s := n - d$, if $v := \lceil \log(q) \cdot s \rceil \leq k - 1$ then $\mathsf{F} \colon \{0,1\}^{k+\ell} \to \{0,1\}^v$ defined so that on input $(m, r) \in \{0,1\}^{k+\ell}$ it computes $\mathrm{Enc}_r(m)$ and outputs a binary representation of the first s symbols is a distributional one-way function.[9] [10]

The proof for the lemma is given below, but first, we will use it to prove Theorem 24.

Proof of Theorem 24. Let $d = Max(\lceil \rho_0 \cdot n \rceil, \lceil 2\rho_1 \cdot n \rceil)$ and note that by definition at least one of the following holds: The code detects d-errors, or corrects $\lceil d/2 \rceil$-errors. Moreover, by our assumption on the rate of the code, it holds that $d > n - R \cdot n + 1$. Let $s = n - d$ and let $v = \lceil \log(q) \cdot s \rceil$ since $R \geq \frac{k}{\log(q) \cdot n}$ it holds that $v \leq k - 1$. Thus, we meet the conditions of Lemma 25 which implies that there exists a distributional one-way function (in fact outputting the binary representation of the first s symbols of Enc is the distributional one-way function), and by Theorem 7 there exists a one-way function. ◀

Proof of Lemma 25. For $k \in \mathbb{N}$, let M_k and R_k denote the random variables sampled uniformly from $\{0,1\}^k$ and $\{0,1\}^{\ell(k)}$ respectively. Let $Y_k = \mathsf{F}(M_k, R_k)$ and assume for contradiction that F is not a distributional one-way function (see Definition 6). That is, for every constant $c > 0$ there exists a PPT algorithm A such that for infinitely many $k \in \mathbb{N}$,

$$\mathsf{SD}\Big(\big(\mathsf{A}(Y_k), Y_k\big), \big((M_k, R_k), Y_k\big)\Big) \leq 1/k^c. \tag{1}$$

For a fixed k for which the above holds, we omit the subscript k from the random variables to avoid clutter.

Let $(M', R') = \mathsf{A}(Y)$ and note that by the definition of average min-entropy (see Definition 3)

$$\mathbf{Pr}\left[M = M'\right] \leq 2^{-\tilde{H}_\infty(M|Y)} \leq 1/2 \tag{2}$$

where the last inequality follows by Lemma 4 that states that $\tilde{H}_\infty(M|Y) \geq H_\infty(M, Y) - v \geq k - v \geq 1$.

Let $C = \mathrm{Enc}_R(M)$ and $C' = \mathrm{Enc}_{R'}(M')$ and recall that by definition, $Y = \mathsf{F}(M, R)$ is just the binary representation of the first s symbols of C. Thus, if $\mathsf{F}(M', R') = \mathsf{F}(M, R) = Y$ it follows that, $\Delta(\mathrm{Enc}_R(M), \mathrm{Enc}_{R'}(M')) \leq n - s = d$, which by Equation 1 implies that:

$$\mathbf{Pr}\left[\Delta(C, C') > d\right] \leq 1/k^c. \tag{3}$$

Note also that by Equation 1,

$$\mathbf{Pr}\left[M' = \perp\right] \leq 1/k^c \tag{4}$$

We first handle the case where the code detects d-error and then the case where the code corrects $\lceil d/2 \rceil$-errors.

[9] Here, ℓ is just the implicitly assumed bound on the number of random bits used by Enc.

[10] The statement of Lemma 25 could be readily extended so that restricting the output of $\mathrm{Enc}_r(m)$ to *any* s symbols (that can be efficiently identified) not just the first s symbols is itself a distributional one-way function. This would be similar to our statement on self-seeded codes made in the full version [22], where we show that restricting the output to any subset of size s is a collision resistance hash family. However, for the sake of simplicity, we restrict our statement and analysis to the simple case where the distributional one-way function is defined to be the first s symbols.

Detection. By Equation 1 and by the reliability of the code,

$$\mathbf{Pr}\left[\mathrm{Dec}(C') = M'\right] \geq \mathbf{Pr}\left[\mathrm{Dec}(C) = M\right] - 1/k^c > 0.9 \tag{5}$$

We now define a PPT algorithm A' (that makes use of A) that contradicts the detection property of the code: On input 1^k, A' samples $m \leftarrow \{0,1\}^k$ and on input $c \in [q]^n$ it runs $(m', r') \leftarrow \mathsf{A}(y)$ where y is just the binary representation of the first s symbols of c (i.e., $y = \mathsf{F}(m, r)$), and outputs $c' = \mathrm{Enc}_{r'}(m')$. It holds that:

- $M = \mathsf{A}'(1^k)$ (where by definition M is sampled uniformly by A').
- Compute $C = \mathrm{Enc}_R(M)$ (by sampling $R \leftarrow \{0,1\}^\ell$).
- $C' = \mathrm{Enc}_{R'}(M') = \mathsf{A}'(M, C)$.

Thus, A' contradicts the d-detection property since

$$\begin{aligned}
&\mathbf{Pr}\left[\mathrm{Dec}(C') \notin \{M, \perp\} \wedge \Delta(C', C) \leq d\right] \\
&\geq \mathbf{Pr}\left[\mathrm{Dec}(C') = M' \wedge M' \notin \{M, \perp\} \wedge \Delta(C', C) \leq d\right] \\
&\geq \mathbf{Pr}\left[\mathrm{Dec}(C') = M'\right] - \mathbf{Pr}\left[M' = M\right] - \mathbf{Pr}\left[M' = \perp\right] - \mathbf{Pr}\left[\Delta(C', C) > d\right] \\
&\geq 0.9 - 0.5 - 2/k^c \geq 0.1
\end{aligned}$$

Where the penultimate inequality follows from Equations 5, 2, 4 and 3.

Correction. We now prove the correction property. Let G be a function defined as follows: On inputs $c, c' \in [q]^n$, G samples a set I of random $(n-s)/2$ indices from the set $[n] \setminus [s]$ (if $n - s$ is an odd number then with probability $1/2$ it sample $\lceil (n-s)/2 \rceil$ indices and with probability $1/2$ it samples $\lceil (n-s)/2 \rceil - 1$ indices), and outputs $c^* \in [q]^n$ where for every $i \in I$, $c^*_i = c'_i$ and for every $i \in [n] \setminus I$, $c^*_i = c_i$. Intuitively, if c and c' agree on the first s indexes then $G(c, c') = c^*$ is a random "mid point" between c and c'.

Recall that $C = \mathrm{Enc}_R(M)$, $C' = \mathrm{Enc}_{R'}(M')$ and let $C^* = G(C, C')$. By Equation 1 and symmetry,

$$\mathsf{SD}((M, R, C, C^*), (M', R', C', C^*)) \leq 2/k^c. \tag{6}$$

Since by assumption $n - s = d$, by the definition of C^* it follows that:

$$\mathbf{Pr}\left[\Delta(C, C^*) > \lceil d/2 \rceil\right] \leq 1/k^c \tag{7}$$

Recall that (M, R) are sampled uniformly at random, thus, by Equations 6 and 7 and the decoding properly of the code

$$\mathbf{Pr}\left[\mathrm{Dec}(C^*) = M'\right] \geq \mathbf{Pr}\left[\mathrm{Dec}(C^*) = M\right] - 3/k^c \geq 0.9. \tag{8}$$

We now define a PPT algorithm A' (that makes use of A) that contradicts the correction property of the code: On input 1^k, A' samples $m \leftarrow \{0,1\}^k$ and on input $c \in [q]^n$ runs $(m', r') \leftarrow \mathsf{A}(y)$ where y is just the binary representation of the first s symbols of c (i.e., $y = \mathsf{F}(m, r)$), computes $c' = \mathrm{Enc}_{r'}(m')$ and outputs $c^* = G(c, c')$. It holds that,

- $M = \mathsf{A}'(1^k)$ (where by definition M is sampled uniformly by A').
- Compute $C = \mathrm{Enc}_R(M)$ (by sampling $R \leftarrow \{0,1\}^\ell$).
- $C^* = \mathsf{A}'(M, C)$ (recall that $C^* = G(C, C')$ for $C' = \mathrm{Enc}_{R'}(M')$ and $(M', R') = \mathsf{A}(\mathsf{F}(M, R))$).

Thus, A' contradicts the $\lceil d/2 \rceil$-correction property since,

$$\begin{aligned}
&\mathbf{Pr}\left[\mathrm{Dec}(C^*) \neq M \wedge \Delta(C^*, C) \leq \lceil d/2 \rceil\right] \\
&\geq \mathbf{Pr}\left[\mathrm{Dec}(C^*) = M' \wedge M \neq M' \wedge \Delta(C^*, C) \leq \lceil d/2 \rceil\right] \\
&\geq \mathbf{Pr}\left[\mathrm{Dec}(C^*) = M'\right] - \mathbf{Pr}\left[M = M'\right] - \mathbf{Pr}\left[\Delta(C^*, C) \leq \lceil d/2 \rceil\right] \\
&\geq 0.9 - 0.5 - 1/k^c \geq 0.1.
\end{aligned}$$

Where the penultimate inequality follows from Equations 2, 7 and 8. This concludes the proof for Lemma 25. ◀

6 Self-Seeded Codes Imply Collision Resistance Hash Functions

We now show that non-trivial self-seeded codes imply the existence of a collision resistance hash family. Loosely speaking, the following theorem states that if a self-seeded code has rate that bypasses the Singleton bound, then it implies collision resistance hash functions (since there are codes with large alphabet that can achieve the Singleton bound, this result means that for large alphabet, any non-trivial self-seeded code implies collision resistance functions).

We note that while the result below is stated with respect to the notion of self-seeded codes, the same exact proof also holds for the weaker notion seeded codes defined in [3] in which the encoding and decoding algorithms both share the random seed. Since our lower bound still holds even if the decoding algorithm is given the random seed as input, the lower bound immediately extends to the weaker notion of seeded codes. However, to avoid confusion and clutter, we present our result with respect to the notion of self-seeded codes defined in this paper.

▶ **Theorem 26** (Self-Seeded codes with rate beating the Singleton bound imply collision resistance hash functions)**.** *Let* $(\mathrm{Enc}, \mathrm{Dec})$ *be a self-seeded code with rate* $0 \leq R < 1$*. If one of the following holds:*

- *The code detects* ρ_0*-relative errors and* $\rho_0 > 1 - R + \frac{1}{n}$*.*
- *The code corrects from* ρ_1*-relative errors and* $2\rho_1 > 1 - R + \frac{2}{n}$*.*

Then there exists an explicit construction for a collision resistance hash family given black-box access to the code.

The proof for Theorem 26 appears in the full version [22].

References

1 Jeremiah Blocki, Venkata Gandikota, Elena Grigorescu, and Samson Zhou. Relaxed locally correctable codes in computationally bounded channels. *IEEE Transactions on Information Theory*, 67(7):4338–4360, 2021. doi:10.1109/TIT.2021.3076396.

2 Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM journal on computing*, 38(1):97–139, 2008. doi:10.1137/060651380.

3 Ofer Grossman, Justin Holmgren, and Eylon Yogev. Transparent error correcting in a computationally bounded world. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part III*, volume 12552 of *Lecture Notes in Computer Science*, pages 530–549. Springer, 2020. doi:10.1007/978-3-030-64381-2_19.

4 V. Guruswami and A. Smith. Optimal rate code constructions for computationally simple channels. *Journal of the ACM (JACM)*, 63(4):35, 2016.

5 Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Transactions on information theory*, 54(1):135–150, 2008. doi:10.1109/TIT.2007.911222.

6 Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. Essential coding theory, 2012. Draft available at https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/web-coding-book.pdf.

7 R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, 1950. doi:10.1002/j.1538-7305.1950.tb00463.x.

8 Brett Hemenway, Rafail Ostrovsky, Martin J. Strauss, and Mary Wootters. Public key locally decodable codes with short keys. In Leslie Ann Goldberg, Klaus Jansen, R. Ravi, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 14th International Workshop, APPROX 2011, and 15th International Workshop, RANDOM 2011, Princeton, NJ, USA, August 17-19, 2011. Proceedings*, volume 6845 of *Lecture Notes in Computer Science*, pages 605–615. Springer, 2011. doi:10.1007/978-3-642-22935-0_51.

9 Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography. In *FOCS 30*, pages 230–235, 1989. doi:10.1109/SFCS.1989.63483.

10 Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. High-rate locally correctable and locally testable codes with sub-polynomial query complexity. *Journal of the ACM (JACM)*, 64(2):1–42, 2017. doi:10.1145/3051093.

11 R. J. Lipton. A new approach to information theory. In *11th Annual Symposium on Theoretical Aspects of Computer Science*, pages 699–708, 1994. doi:10.1007/3-540-57785-8_183.

12 Silvio Micali, Chris Peikert, Madhu Sudan, and David A. Wilson. Optimal error correction for computationally bounded noise. *IEEE Trans. Information Theory*, 56(11):5673–5680, 2010. doi:10.1109/TIT.2010.2070370.

13 Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*, pages 33–43. ACM, 1989. doi:10.1145/73007.73011.

14 Rafail Ostrovsky, Omkant Pandey, and Amit Sahai. Private locally decodable codes. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *Automata, Languages and Programming, 34th International Colloquium, ICALP 2007, Wroclaw, Poland, July 9-13, 2007, Proceedings*, volume 4596 of *Lecture Notes in Computer Science*, pages 387–398. Springer, 2007. doi:10.1007/978-3-540-73420-8_35.

15 Leonid Reyzin. Some notions of entropy for cryptography: (invited talk). In *International Conference on Information Theoretic Security*, pages 138–142. Springer, 2011. doi:10.1007/978-3-642-20728-0_13.

16 John Rompel. One-way functions are necessary and sufficient for secure signatures. In Harriet Ortiz, editor, *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 387–394. ACM, 1990. doi:10.1145/100216.100269.

17 R. Shaltiel and J. Silbak. Explicit list-decodable codes with optimal rate for computationally bounded channels. *Comput. Complex.*, 30(1):3, 2021. doi:10.1007/s00037-020-00203-w.

18 R. Shaltiel and J. Silbak. Explicit uniquely decodable codes for space bounded channels that achieve list-decoding capacity. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1516–1526, 2021. doi:10.1145/3406325.3451048.

19 R. Shaltiel and J. Silbak. Error correcting codes that achieve BSC capacity against channels that are poly-size circuits. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 13–23, 2022. doi:10.1109/FOCS54457.2022.00009.

20 R. Shaltiel and J. Silbak. Explicit codes for poly-size circuits and functions that are hard to sample on low entropy distributions. *Electronic Colloquium on Computational Complexity (ECCC)*, 2024. URL: https://eccc.weizmann.ac.il/report/2023/149.

21 Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(4):623–656, 1948. doi:10.1002/J.1538-7305.1948.TB00917.X.

22 Jad Silbak and Daniel Wichs. Detecting and correcting computationally bounded errors: A simple construction under minimal assumptions. Cryptology ePrint Archive, Paper 2024/1461, 2024. URL: https://eprint.iacr.org/2024/1461.

Optimal Communication Complexity of Chained Index

Janani Sundaresan
Cheriton School of Computer Science, University of Waterloo, Canada

Abstract

We study the CHAIN communication problem introduced by Cormode et al. [ICALP 2019]. For $k \geq 1$, in the $\text{CHAIN}_{n,k}$ problem, there are k string and index pairs (X_i, σ_i) for $i \in [k]$ such that the value at position σ_i in string X_i is the same bit for all k pairs. The input is shared between $k+1$ players as follows. Player 1 has the first string $X_1 \in \{0,1\}^n$, player 2 has the first index $\sigma_1 \in [n]$ and the second string $X_2 \in \{0,1\}^n$, player 3 has the second index $\sigma_2 \in [n]$ along with the third string $X_3 \in \{0,1\}^n$, and so on. Player $k+1$ has the last index $\sigma_k \in [n]$. The communication is one way from each player to the next, starting from player 1 to player 2, then from player 2 to player 3 and so on. Player $k+1$, after receiving the message from player k, has to output a single bit which is the value at position σ_i in X_i for any $i \in [k]$. It is a generalization of the well-studied INDEX problem, which is equivalent to $\text{CHAIN}_{n,2}$.

Cormode et al. proved that the $\text{CHAIN}_{n,k}$ problem requires $\Omega(n/k^2)$ communication, and they used it to prove streaming lower bounds for the approximation of maximum independent sets. Subsequently, Feldman et al. [STOC 2020] used it to prove lower bounds for streaming submodular maximization. However, it is not known whether the $\Omega(n/k^2)$ lower bound used in these works is optimal for the problem, and in fact, it was conjectured by Cormode et al. that $\Omega(n)$ bits are necessary.

We prove the optimal lower bound of $\Omega(n)$ for $\text{CHAIN}_{n,k}$ when $k = o(n/\log n)$ as our main result. This settles the open conjecture of Cormode et al., barring the range of $k = \Omega(n/\log n)$. The main technique is a reduction to a non-standard INDEX problem where the input to the players is such that the answer is biased away from uniform. This biased version of INDEX is analyzed using tools from information theory. As a corollary, we get an improved lower bound for approximation of maximum independent set in vertex arrival streams via a reduction from CHAIN directly.

2012 ACM Subject Classification Theory of computation → Communication complexity

Keywords and phrases communication complexity, index communciation problem

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.89

Related Version *Full Version*: https://arxiv.org/abs/2404.07026

Funding *Janani Sundaresan*: Supported in part by Sepehr Assadi's Sloan Research Fellowship and startup grant from University of Waterloo.

Acknowledgements The author is thankful to Sepehr Assadi for introducing them to the problem and for insightful discussions on the proof. The author would also like to thank Parth Mittal for useful comments, Christian Konrad for introducing them to the Augmented Chain problem, and the anonymous reviewers of ITCS 2025 for helpful comments and suggestions. The author is very grateful to Mi-Ying Huang, Xinyu Mao, Guangxu Yang and Jiapeng Zhang for an illuminating discussion about the problem. They pointed out an important flaw in an earlier version of this work, and the discussion was instrumental for the new proofs in the current version.

1 Introduction

The INDEX problem is one of the foundational problems in communication complexity. For $n \geq 1$, in the INDEX_n problem, there are two players Alice and Bob. Alice has a string $X \in \{0,1\}^n$ and Bob has an index $\sigma \in [n]$, and Bob has to output the value of X at position σ.


© Janani Sundaresan;
licensed under Creative Commons License CC-BY 4.0

If the communication is one-way from Alice to Bob, it is easy to show that Alice needs to send $\Omega(n)$ bits to get any constant advantage [1, 30]. This problem has been well-studied in multiple settings, and we know tight trade-offs in the two-party communication model for communication complexity [34], information complexity [24], and quantum communication complexity [8, 24].

Among the numerous applications of communication complexity, one that is of interest to us is proving lower bounds for streaming algorithms. INDEX and its variants, in particular, have been quite useful in this context, for example, in [23, 18, 20, 21, 16, 10]. This is by no means an exhaustive list.

In this paper, we study a natural generalization of INDEX, called chained index ($\text{CHAIN}_{n,k}$ for $n, k \geq 1$) introduced by [12]. There are k different instances of INDEX_n, correlated so that they have the same answer. They are "chained" together, where each player holds the index to the previous instance, and also the string for the next instance.

▶ **Definition 1** (Informal). *In $\text{CHAIN}_{n,k}$, there are k instances of INDEX_n, all with the same answer. Players 1 and 2 take on the role of Alice and Bob respectively in the first instance, players 2 and 3 take on the role of Alice and Bob respectively for the second instance, and so on, all the way till players k and $k+1$ for the last instance.*

Communication is one-way from each player to the next in ascending order. The last player has to output the answer. The communication cost is the total number of bits in all the messages sent by the players. See Figure 1 for an illustration.

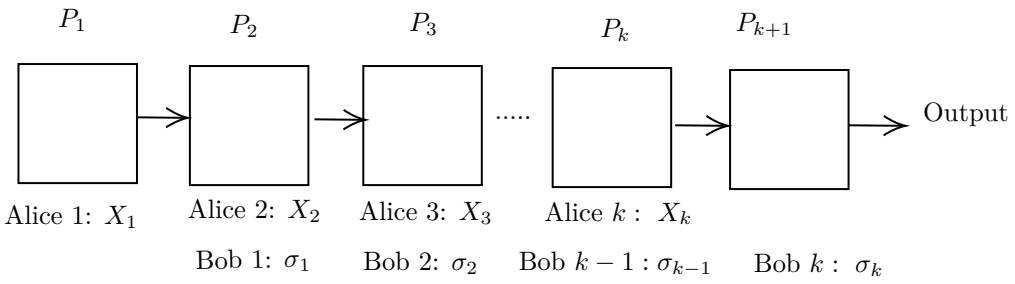


Figure 1 An illustration of the $\text{CHAIN}_{n,k}$ problem with k correlated sub-instances of INDEX_n from Definition 1. The arrows illustrate that the message is from P_i to P_{i+1} for $i \in [k]$.

In [12], a reduction from CHAIN was employed to get a lower bound for approximation of maximum independent sets in vertex arrival streams. Before its introduction, [28] used the problem implicitly to get a lower bound of $(1 - 1/e)$ in the approximation factor for maximum matching in $\tilde{O}(n)$ space in vertex arrival streams.

The problem has been used by the breakthrough result of [19] to study the multi-party communication complexity of submodular maximization. They proved that any randomized p-party protocol which maximizes a monotone submodular function $f : \{0,1\}^N \to \mathbb{R}$, subject to a cardinality constraint of at most p and an approximation factor of at least $(1/2 + \epsilon)$, uses $\Omega(N\epsilon/p^3)$ communication. This also gave a lower bound for streaming submodular maximization. The CHAIN problem was used by [17] also for similar purposes, but subject to stronger matroid constraints. [7] used a reduction from CHAIN to prove lower bounds for interval independent set selection in streams of split intervals.

We do not know tight bounds for the communication complexity of the CHAIN problem, despite finding varied applications of it. There is a trivial protocol of $O(n)$ bits, where any player can send the entire string to the next player who holds the index. Another simple

protocol is for each player to send $O(n/k)$ bits randomly sampled from their strings using public randomness, and with constant probability, in at least one of the k instances, we send the special bit to the player holding the index. However, this still takes $\Omega(n)$ total bits of communication.

In [12], they prove a lower bound of $\Omega(n/k^2)$ for any $k \geq 1$ through a reduction from conservative multi-party pointer jumping problem, introduced by [14]. They state without proof that a stronger lower bound of $\Omega(n/k)$ can be obtained for a restricted range of $k \leq O((n/\log n)^{1/4})$. They posed the following conjecture on the optimal communication lower bound.

▶ **Conjecture 2 ([12]).** *Any protocol that solves* $\textsc{chain}_{n,k}$ *requires* $\Omega(n)$ *bits of communication.*

[19] made some progress on Conjecture 2 by showing that among all the messages sent by the players, there is at least one message with $\Omega(n/k^2)$ bits for every $k \geq 1$. But the original conjecture is still open, and this is the focus of our work.

1.1 Our Results

We settle Conjecture 2 almost fully by proving the optimal lower bound of $\Omega(n)$ barring the corner case of when k is too large. As far as we know, this corner case is not a focus for existing reductions from $\textsc{chain}_{n,k}$.

▶ **Theorem 3.** *For any* $n, k \geq 1$*, any protocol for* $\textsc{chain}_{n,k}$ *with probability of success at least 2/3, requires* $\Omega(n - k\log n)$ *total bits of communication.*

Therefore, as long as $k = o(n/\log n)$, we get the optimal $\Omega(n)$ lower bound from Theorem 3.

The proof of Theorem 3 can be found in Section 3. We prove the lower bound in the more general blackboard model of communication, instead of private messages between players (see Section 2.1 for details). The main idea is to analyze $\textsc{index}_n$ where Alice and Bob already have *some prior advantage* in guessing the answer. We elaborate on our techniques in Section 1.2.

As a direct corollary of Theorem 3, we get improvements in streaming lower bounds in [12, 19] through reductions from CHAIN immediately. In particular, we get that any algorithm which α-approximates the size of a maximum independent set in vertex arrival streams requires $\Omega(n^2/\alpha^5 - \log n)$ space, while the previous bound was $\Omega(n^2/\alpha^7)$ in [12]. We present the implications of our result in Section 4.

A further generalization of CHAIN, called Augmented Chain was defined in [15]. Here, instances of Augmented Index are chained together instead. Our lower bound of $\Omega(n - k\log n)$ can be extended to Augmented Chain also, and the details are covered in Section 3.4.

1.2 Our Techniques

In this subsection, we give an overview of the challenges in proving the lower bound and a summary of our techniques. We start by going over the prior techniques.

Prior Techniques

We will briefly talk about the technique used in [19] to prove that there is at least one player who sends $\Omega(n/k^2)$ bits for any $k \geq 1$. We will argue that these techniques can be extended to proving a lower bound of $\Omega(n/k)$ for the total number of bits, but not all the way to $\Omega(n)$ bits.

The first step in proving a lower bound for $\text{CHAIN}_{n,k}$ in [19] is a decorrelation step – the k instances of INDEX_n have the same answer, and they remove this correlation with a hybrid argument. These arguments have been used extensively in the literature (see e.g., [26, 3, 27, 6]). Intuitively, any protocol that tries to solve $\text{CHAIN}_{n,k}$ may attempt to solve "many" of the instances of INDEX_n, albeit each with a "small" advantage over $1/2$, in the hope that the "small" advantages may accrue to get a constant probability of success overall (taking advantage of the fact that all the instances of INDEX_n have the same answer). The hybrid argument is a way to reduce proving a lower bound on the overall problem, to proving a lower bound for k different INDEX_n problems against these low advantages.

Let us assume, for simplicity, that the protocol tries to get an advantage of $\Omega(1/k)$ in each instance of INDEX_n, to get a constant total advantage. We can prove that any protocol that gets an advantage of $\Omega(1/k)$ for INDEX_n uses $\Omega(n/k^2)$ bits of communication using basic tools from information theory [9] (this is quite standard, see e.g., [2] for a direct proof), and this is known to be tight. Therefore, for k instances, we get a lower bound of $\Omega(n/k)$ bits in total. Now, we will argue why this is not the optimal lower bound.

On one hand, for INDEX_n, it is known that for any $\delta \in (0, 1/2)$, there is a protocol that uses $O(n\delta^2)$ bits of communication to get a probability of success $1/2 + \delta$. This means that INDEX_n can be solved with advantage $\Omega(1/k)$ in $O(n/k^2)$ bits; in other words, each "hybrid step" of the previous lower bound argument is optimal. So, then, why can we not get a good protocol for $\text{CHAIN}_{n,k}$ by running the protocol for INDEX_n with $\delta = 1/k$ on all k instances? This protocol would have $O(n/k)$ bits of communication in total. The reason is that the $1/k$ small advantages do not add up as we would like them to. To illustrate this, we will briefly talk about the protocol that gets δ advantage in $O(n\delta^2)$ bits for INDEX_n.

Protocol for INDEX_n

First, we will sketch a protocol for $\delta = 1/\sqrt{n}$ that uses $O(1)$ bits of communication. Let us imagine that the input X is chosen uniformly at random from $\{0,1\}^n$ and the index σ is chosen uniformly at random from $[n]$. Then, Alice finds the majority bit from her string X and sends it to Bob. Bob just outputs the bit sent by Alice. We know, from simple anti-concentration bounds on the binomial distribution, that the number of indices with the majority bit in X is at least $n/2 + c\sqrt{n}$ with constant probability for some appropriate constant c. The protocol succeeds if Bob holds the index σ to a majority bit, and this happens with probability at least $\frac{1}{2} + \frac{c}{\sqrt{n}}$.

The assumption that the input X is chosen uniformly at random from $\{0,1\}^n$ can be removed using public randomness. Alice and Bob collectively sample a random string $A \in \{0,1\}^n$, and Alice changes her input to $X \oplus A$ so that each bit is 0 or 1 with equal probability. Similarly, the assumption that the index σ is chosen uniformly at random from $[n]$ can be removed by Alice and Bob sampling a random permutation of $[n]$. Alice permutes string X according to this permutation, and Bob changes his input to the index that the permutation maps σ to. This is termed as the self-reducibility property of INDEX.

We can also extend the protocol to any δ by partitioning the string X into $n\delta^2$ blocks at random using public randomness and sending the majority bit in each block. Bob knows the block that his input index σ belongs to as public randomness is used.

Challenges for $\text{CHAIN}_{n,k}$

If we use the protocol we described for $\text{CHAIN}_{n,k}$, we are left with k bits from each instance of INDEX_n, which may be the correct answer to the problem with probability $1/2 + \Theta(1/k)$, but, the variance of each of these bits is $1/4 - \Theta(1/k^2)$. We have a protocol for $\text{CHAIN}_{n,k}$,

where, out of the k instances of INDEX_n, it finds the right answer for $\approx k/2 + \Theta(1)$ instances in expectation. However, the standard deviation of the number of right answers is $\approx \sqrt{k}/2$, which is enough to mask the $\Theta(1)$ improvement over $k/2$ we get in expectation. If we use a hybrid argument over the total variation distance, each of the smaller "hybrid steps" is optimal, whereas the overall lower bound is not optimal. We cannot achieve a lower bound stronger than $\Omega(n/k)$.

Our Solution

Instead of keeping track of progress in terms of advantage gained in guessing the answer, we directly keep track of the "information" the message reveals about the answer. Formally, this translates to the change in entropy of the answer, after each successive message. Initially, the players have no information about the answer, and it is uniform over $\{0,1\}$ (the entropy is 1). Any protocol with a large enough probability of success must reduce the entropy of the answer by a large factor (see Fano's inequality in Proposition 8). We prove that after each message, the entropy is reduced only by an additive factor *linear in the length of the message*, by a reduction to the INDEX problem. This will give us a lower bound on the total length of the messages.

For INDEX_n, in any protocol with probability of success at least ε, the entropy of the answer conditioned on the message is at most $H_2(\varepsilon)$ where $H_2(x) = x\log(1/x)+(1-x)\log(1/(1-x))$ is the binary entropy function. In the standard version where the initial entropy is 1, the reduction in entropy is $1 - H_2(\varepsilon)$, and it is known that the protocol requires $\Omega(n(1 - H_2(\varepsilon)))$ communication [24]. This is not sufficient for our application due to the following reason: after the message of $\mathcal{P}_1$, when $\mathcal{P}_2$ and $\mathcal{P}_3$ attempt to solve INDEX_n, they already have some prior advantage that the message of $\mathcal{P}_1$ gives them. Therefore, we need to analyze INDEX_n when the answer is not uniform over $\{0,1\}$.

Biased Index

We define the biased index problem, parametrized by $\theta \in [-1/2, 1/2]$. Alice and Bob receive input $Y \in \{0,1\}^n$ and $\rho \in [n]$ respectively, such that the value at position ρ in Y (denoted by $Y(\rho)$) is 1 with probability $1/2 + \theta$ and 0 otherwise. Alice sends a message M to Bob and Bob has to output $Y(\rho)$. The initial entropy of the answer is $H_2(1/2 + \theta)$. We prove that entropy of $Y(\rho)$ conditioned on M is smaller by at most $O((|M| + \log n)/n)$ compared to the initial $H_2(1/2 + \theta)$, which is **our main contribution** (see Lemma 14).

We can show that the entropy of input to Alice, i.e. the random string Y, in such a distribution is at least $\Omega(n \cdot H_2(1/2 + \theta))$. Hence, after a message of length s from Alice, the entropy of string Y reduces to $\Omega(n \cdot H_2(1/2 + \theta) - s)$. For a randomly chosen position in Y after conditioning on the message, the entropy is at least $\approx H_2(1/2 + \theta) - s/n$, which gives a lower bound on s.

In the distribution given to Alice and Bob, however, ρ is *not* chosen uniformly at random, and in fact, is correlated with the distribution of Y. Such versions of index where the distributions of Alice and Bob are correlated have been studied before (see Sparse Indexing in Appendix A of [5] and Section 3.3 of [36]). This correlation is the main issue in analyzing biased index with information theoretic tools.

Adapted from the approach in Appendix A of [5], we restrict the randomness in Y to a fixed set of indices of a carefully chosen size (based on θ), and break this correlation. The loss in entropy of Y is not significant enough to hinder us, and the restriction then allows us to use standard information theoretic tools to analyze biased index (see Section 3.3 for more details).

Independent and Concurrent Work

Independently and concurrently of this work, [33] made progress on Conjecture 2. They showed a lower bound of $\Omega(n/k + \sqrt{n})$ for oblivious protocols (where the length of the message sent by each player does not depend on the input), and a lower bound of $\Omega(n/k - k)$ for general protocols. We show a lower bound of $\Omega(n - k \log n)$ for all protocols.[1]

Quantitatively, our lower bounds are a factor of almost k stronger, and are optimal for $k = o(n/\log n)$; moreover, our lower bound also holds for the Augmented Chain problem of [15]. In terms of techniques, however, the two works are entirely disjoint: their proof is based on a new method of analysis through min-entropy and we use information theoretic approaches.

2 Preliminaries

In this section, we will present the required notation and definitions for our proof.

Notation

For any tuple $A = (A_1, A_2, \ldots, A_m)$ of m items, we use $A_{<i}$ to denote the tuple $(A_1, A_2, \ldots, A_{i-1})$ for all $i \in [m]$. We use sans-serif font to denote random variables. For any random variable A, we use $A \sim \mathsf{A}$ to denote any A sampled from the distribution of the random variable A.

For any string $X \in \{0,1\}^n$, we use $X(\sigma)$ to denote the bit at position σ in X for $\sigma \in [n]$. We use $X(< \sigma)$ to denote the string of $\sigma - 1$ bits preceding $X(\sigma)$ in X. We use $X(S)$ for any $S \subseteq [n]$ to denote the bits at positions in set S.

For any $x \in [0,1]$, we use $H_2 : [0,1] \to [0,1]$ to denote the binary entropy function.

$$H_2(x) = -x \log x - (1-x) \log(1-x).$$

We need the following standard approximation of binomial coefficients (see Lemma 7 of Chapter 10 in [32]).

▶ **Fact 4** (c.f. [32]). *For any $p \geq 1$ and any $q \in [p-1]$, we have,*

$$2^{p \cdot H_2(q/p)} \cdot \sqrt{\frac{n}{8\pi q(p-q)}} \leq \binom{p}{q} \leq 2^{p \cdot H_2(q/p)} \cdot \sqrt{\frac{n}{2\pi q(p-q)}}$$

2.1 Communication Complexity Model

We use the standard number-in-hand multi-party model of communication. Only the basic definitions are given in this subsection. More details can be found in textbooks on communication complexity [35, 31].

For any $k \geq 1$, let f be a function from $\mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_k$ to $\{0,1\}$. There are k players $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_k$ where $\mathcal{P}_i$ gets an input $a_i \in \mathcal{A}_i$ for $i \in [k]$. There is a shared blackboard visible to all the players. The players have access to a shared tape of random bits, along with their own private randomness. In any protocol π for f, the players send a message to the blackboard in increasing order ($\mathcal{P}_1$ sends a message followed by $\mathcal{P}_2$, and so on till

[1] The authors of [33] pointed out an important flaw in the arguments of an earlier version of this work which was posted at around the same time as [33]. This flaw was subsequently fixed in the current version using a global change to the original argument, which now recovers optimal result for $k = o(n/\log n)$.

$\mathcal{P}_k$). The last player $\mathcal{P}_k$, after all the messages are sent, outputs a single bit denoted by $\pi(a_1, a_2, \ldots, a_k)$. Protocol π is said to solve f with probability of success at least $1-\delta$ if, for all $i \in [k]$, for any choice of $a_i \in \mathcal{A}_i$, we have,

$$\Pr[\pi(a_1, a_2, \ldots, a_k) \neq f(a_1, a_2, \ldots, a_k)] \leq \delta.$$

The **communication cost** of a protocol π is defined as the worst case **total communication** of all the players on the blackboard at the end of the protocol.

▶ **Definition 5.** *The* **randomized communication complexity** *of f, with probability of error δ, is defined as the minimum communication cost of any protocol which solves f with probability of success at least $1-\delta$.*

2.2 Information Theoretic Tools

Our proof relies on tools from information theory, and we state the basic definitions and the inequalities we need in this section. Proofs of the statements and more details can be found in Chapter 2 of a textbook on information theory by Cover and Thomas [13].

▶ **Definition 6** (Shannon Entropy)**.** *For any random variable X over support $\mathcal{A}$, the Shannon entropy of X, denoted by $\mathbb{H}(\mathsf{X})$ is defined as,*

$$\mathbb{H}(\mathsf{X}) = \sum_{A \in \mathcal{A}} \Pr[\mathsf{X} = A] \cdot \log(1/\Pr[\mathsf{X} = A]).$$

For any event $\mathcal{E}$ we define $\mathbb{H}(\mathsf{X} \mid \mathcal{E})$ in the same way, as the entropy of distribution of X conditioned on the event $\mathcal{E}$. For any two random variables X and Y, the entropy of X conditioned on Y, denoted by $\mathbb{H}(\mathsf{X} \mid \mathsf{Y})$ is defined as,

$$\mathbb{H}(\mathsf{X} \mid \mathsf{Y}) = \mathop{\mathbb{E}}_{Y \sim \mathsf{Y}} \mathbb{H}(\mathsf{X} \mid \mathsf{Y} = Y).$$

▶ **Fact 7.** *We know the following about entropy and mutual information:*

1. *For any random variable X, the entropy obeys the bound: $0 \leq \mathbb{H}(\mathsf{X}) \leq \log_2(|\mathcal{X}|)$ where $\mathcal{X}$ is the support of X.*
2. *For any two random variables X, Y, $\mathbb{H}(\mathsf{X} \mid \mathsf{Y}) \leq \mathbb{H}(\mathsf{X})$ with equality holding iff $\mathsf{X} \perp \mathsf{Y}$.*
3. *Chain Rule of Entropy: For $m \geq 1$ and any tuple of random variables $\mathsf{X} = (\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_m)$, $\mathbb{H}(\mathsf{X}) = \sum_{i \in [m]} \mathbb{H}(\mathsf{X}_i \mid \mathsf{X}_{<i})$.*
4. *Subadditivity of entropy: For $m \geq 1$ and any tuple of random variables $\mathsf{X} = (\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_m)$, $\mathbb{H}(\mathsf{X}) \leq \sum_{i \in [m]} \mathbb{H}(\mathsf{X}_i)$.*

We also need the following proposition, which relates entropy to the probability of correctness while estimating a random variable.

▶ **Proposition 8** (Fano's inequality)**.** *Given a binary random variable X and an estimator random variable Y and a function g such that $g(Y) = X$ with probability at least $1-\delta$ for $\delta < 1/2$,*

$$\mathbb{H}(\mathsf{X} \mid \mathsf{Y}) \leq H_2(\delta).$$

This concludes our preliminaries section.

3 The Lower Bound

In this section, we will prove our lower bound of $\Omega(n)$ on the communication complexity of the $\text{CHAIN}_{n,k}$ problem for $k = o(n/\log n)$. Let us formally define the $\text{CHAIN}_{n,k}$ communication problem first.

▶ **Definition 9.** *The $\text{CHAIN}_{n,k}$ communication problem is defined as follows. Given $k+1$ players $\mathcal{P}_i$ for $i \in [k+1]$ where,*

- *$\mathcal{P}_i$ has a string $X_i \in \{0,1\}^n$ for each $i \in [k]$, and,*
- *$\mathcal{P}_i$ for $1 < i \leq k+1$ has an index $\sigma_{i-1} \in [n]$,*

such that,

$$X_i(\sigma_i) = z,$$

for some bit $z \in \{0,1\}$. The players have a blackboard visible to all the parties. For $i \in [k]$ in ascending order, $\mathcal{P}_i$ sends a single message M_i, after which the index σ_i is revealed to the blackboard at no cost. $\mathcal{P}_{k+1}$ has to output whether z is 0 or 1. Refer to Figure 2 for an illustration.

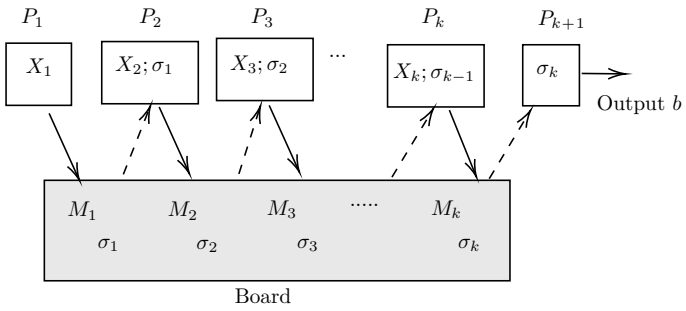


Figure 2 An illustration of the $\text{CHAIN}_{n,k}$ problem from Definition 9. The solid arrows illustrate that player $\mathcal{P}_i$ writes a message M_i to the board. The dashed arrows indicate that $\mathcal{P}_i$ can read the contents of the board. It also shows the order in which the messages are sent by the players and indices are released.

Let us recall the statement of our main result.

▶ **Theorem 3** (restated). *For any $n, k \geq 1$, any protocol for $\text{CHAIN}_{n,k}$ with probability of success at least 2/3, requires $\Omega(n - k\log n)$ total bits of communication.*

We give our hard distribution for $\text{CHAIN}_{n,k}$ in Section 3.1 and give the proof of Theorem 3 in Section 3.2 except for the analysis of biased index, which is given in Section 3.3. Lastly, we extend the arguments to Augmented Chain in Section 3.4.

3.1 Setting Up the Problem

In this subsection, we start by defining the notation for our proof, and we describe the input distributions to $\text{CHAIN}_{n,k}$.

The input hard distribution is as follows. Let $\mathcal{L} \subset \{0,1\}^n$ be the subset of strings where the number of ones is exactly equal to $n/2$.

Distribution $\mathcal{D}$ for $\textsc{chain}_{n,k}$:
1. Pick a bit z uniformly at random from $\{0,1\}$.
2. For each $i \in [k]$, sample (X_i, σ_i) uniformly at random from $\mathcal{L} \times [n]$ and independently conditioned on $X_i(\sigma_i) = z$.

Notation

We use X_i to denote the random variable corresponding to string X_i and $\boldsymbol{\sigma}_i$ to denote the random variable corresponding to index σ_i for $i \in [k]$. To denote the random variable corresponding to the first $i-1$ strings and indices, we use $\mathsf{X}_{<i}$ and $\boldsymbol{\sigma}_{<i}$ respectively.

We use M_i to denote the random variable corresponding to the message M_i sent by $\mathcal{P}_i$ to the blackboard. We use $M = (M_1, M_2, \ldots, M_k)$ to denote the tuple containing the messages of all the players, and M to denote the random variable corresponding to M.

Let π be a deterministic protocol for $\textsc{chain}_{n,k}$ with probability of success at least $2/3$ when the input is distributed according to $\mathcal{D}$. We use Γ to denote the random variable corresponding to the contents of the blackboard (referred to as a **transcript**), and we use γ to also denote transcripts sampled from Γ. We use Γ_i to denote the random variable of the tuple (M^i, σ^i) for $i \in [k]$. We use $\pi(\gamma_{<i}, X_i)$ to denote the output of $\mathcal{P}_i$ when the contents of the blackboard are $\gamma_{<i}$ and the input is X_i for $i \in [k]$.

Let s be the total length of all the messages sent by the players in γ. We assume that the total length of the messages is exactly s by padding. For any random variable A, we use $\mathcal{D}(\mathsf{A})$ to denote the distribution of the random variable A, as the input is distributed according to $\mathcal{D}$. We replace $\mathcal{D}(\mathsf{A} \mid \mathsf{B} = b)$ with $\mathcal{D}(\mathsf{A} \mid b)$ for ease of readability whenever it is clear from context.

Let Z denote the random variable corresponding to bit z. The bit z corresponds to the answer to $\textsc{chain}_{n,k}$. We show the lower bound of $\Omega(n - k \log n)$ for distinguishing between the case when $z = 0$ and $z = 1$ based on the contents of the blackboard.

We need one important observation about the distribution $\mathcal{D}$.

▶ **Observation 10.** *For any $i \in [k]$, random variable $\mathsf{X}_i, \boldsymbol{\sigma}_i$ is independent of $\Gamma_{<i}$ conditioned on Z.*

Proof. For any fixed value of Z, $\mathsf{X}_i, \boldsymbol{\sigma}_i$ are chosen uniformly at random from $\mathcal{L} \times [n]$ such that $\mathsf{X}_i(\boldsymbol{\sigma}_i) = \mathsf{Z}$. This choice is independent of any $\mathsf{X}_j, \boldsymbol{\sigma}_j$ with $j \neq i$, and thus independent of $\Gamma_{<i}$. ◀

We are ready to proceed with the proof of our main theorem.

3.2 Proof of Lower Bound

We start by showing that in any successful protocol, the entropy of the distribution of Z conditioned on the message must be small.

▷ Claim 11 (The transcript reveals information about Z.).

$$\mathbb{H}(\mathsf{Z} \mid \Gamma) \leq 24/25.$$

Proof. We know that $\mathcal{P}_{k+1}$ successfully finds the value of z with probability of success at least $2/3$ using the transcript. Thus, using Fano's inequality in Proposition 8, we have,

$$\mathbb{H}(\mathsf{Z} \mid \Gamma) \leq H_2(2/3) \leq 24/25.$$ ◁

The main part of the proof is that we show a lower bound the entropy of Z conditioned on the transcripts using the entropy of the message.

▶ **Lemma 12.** *For any protocol π,*

$$1 - \frac{12}{n} \cdot (\mathbb{H}(\mathsf{M}) + k \log n) \leq \mathbb{H}(\mathsf{Z} \mid \Gamma).$$

Before we prove Lemma 12, we can easily show that it implies Theorem 3.

Proof of Theorem 3. Combining Lemma 12 with Claim 11, we get,

$$1 - \frac{12}{n} \cdot (\mathbb{H}(\mathsf{M}) + k \log n) \leq \mathbb{H}(\mathsf{Z} \mid \Gamma) \leq \frac{24}{25}.$$

This gives that,

$$\mathbb{H}(\mathsf{M}) \geq \frac{n}{25 \cdot 12} - k \log n.$$

We know from Fact 7-(1) that $\mathbb{H}(\mathsf{M}) \leq \log(2^s) = s$, which proves that the total number of bits $s = \Omega(n - k \log n)$ for any deterministic protocol. By Yao's minimax principle, we get a lower bound of $\Omega(n - k \log n)$ on the randomized communication complexity of $\textsc{chain}_{n,k}$. ◀

The proof of Lemma 12 employs a reduction to the two player $\textsc{index}_n$. However, in these instances of $\textsc{index}_n$, Alice and Bob already have some partial information about the answer. We call this problem the **biased index** problem, and it is defined based on parameter $\theta \in [-1/2, 1/2]$, which is the initial bias known about the answer.

▶ **Definition 13.** *The* **biased index distributional communication problem**, *denoted by* $\textsc{bias-ind}(\theta)$ *for $\theta \in [-1/2, 1/2]$, is defined as follows.*

Sample $W \in \{0, 1\}$ such that $W = 1$ with probability $1/2 + \theta$, and $W = 0$ otherwise. Sample (Y, ρ) uniformly at random from $\mathcal{L} \times [n]$ conditioned on $Y(\rho) = W$. Give string Y to Alice, and index ρ to Bob. Bob has to output $Y(\rho)$ after a single message $M_{\textsc{index}}$ from Alice.

Let $\pi_{\textsc{index}}$ be a deterministic protocol for $\textsc{bias-ind}(\theta)$. Let $\mathsf{W}, \mathsf{Y}, \boldsymbol{\rho}, \mathsf{M}_{\textsc{index}}$ denote the random variables corresponding to W, Y, ρ and $M_{\textsc{index}}$ respectively. Let $\mathcal{D}_\theta$ denote the joint distribution of $\mathsf{W}, \mathsf{Y}, \boldsymbol{\rho}$ and $\mathsf{M}_{\textsc{index}}$ in $\textsc{bias-ind}(\theta)$.

We prove the following lemma about $\textsc{bias-ind}(\theta)$ in Section 3.3.

▶ **Lemma 14** (Biased Index)**.** *For any protocol $\pi_{\textit{INDEX}}$ for $\textsc{bias-ind}(\theta)$,*

$$\mathbb{H}(\mathsf{W} \mid \mathsf{M}_{\textit{INDEX}}, \boldsymbol{\rho}) \geq H_2(1/2 + \theta) - \frac{2}{n} \cdot (\mathbb{H}(\mathsf{M}_{\textit{INDEX}}) + \log n).$$

We can prove Lemma 12 using Lemma 14, but the proof is deferred to the full version.

3.3 Biased Index

In this subsection, we will prove Lemma 14. Let us first recall the input distribution to $\textsc{bias-ind}(\theta)$.

Distribution $\mathcal{D}_\theta$: Sample $W = 1$ with probability $1/2 + \theta$ and set $W = 0$ otherwise. Sample $(Y, \rho) \in \mathcal{L} \times [n]$ conditioned on $Y(\rho) = W$.

In $\mathcal{D}_\theta$, the distribution of Y and $\boldsymbol{\rho}$ are highly correlated. We give an alternate way of sampling Y, ρ so that this correlation is removed partially.

Distribution $\mathcal{D}'_\theta$:
For $\theta \geq 0$:
1. Sample set $T \subset [n]$ of size $b = n/(1+2\theta)$ uniformly at random.
2. Sample a set S of $n/2$ indices from T uniformly at random and set them to 1, and set $[n] \setminus S$ to 0 to get Y.
3. Sample ρ by sampling an index uniformly at random from T.

For $\theta < 0$:
1. Sample set $T \subset [n]$ of size $b = n/(1-2\theta)$ uniformly at random.
2. Sample a set S of $n/2$ indices from T uniformly at random and set them to 0, and set $[n] \setminus S$ to 1 to get Y.
3. Sample ρ by sampling an index uniformly at random from T.

In this section, we assume that $\theta \geq 0$. For the case when $\theta < 0$, the proof follows in the same vein, and is not presented. Let T, S denote the random variables corresponding to set T and set S respectively. We show that distributions $\mathcal{D}_\theta$ and $\mathcal{D}'_\theta$ are in fact identical, and the proofs can be found in the full version.

▷ Claim 15. In distribution $\mathcal{D}_\theta$, for any $(Y, \rho) \in \mathcal{L} \times [n]$,

$$\Pr(\mathsf{Y} = Y, \boldsymbol{\rho} = \rho) = \begin{cases} \frac{1+2\theta}{n \cdot \binom{n}{n/2}} & \text{when } Y(\rho) = 1, \\ \frac{1-2\theta}{n \cdot \binom{n}{n/2}} & \text{when } Y(\rho) = 0. \end{cases}$$

▷ Claim 16. Distribution $\mathcal{D}_\theta$ is the same as $\mathcal{D}'_\theta$.

Using this alternate way of sampling, it is easy to see that random variables Y and $\boldsymbol{\rho}$ are independent of each other conditioned on T. It can also be extended to include random variable $\mathsf{M}_{\text{INDEX}}$, as it is only a function of Y.

▶ **Observation 17.** *In distribution $\mathcal{D}'_\theta$, conditioned on $\mathsf{T} = T$, for any $i \in T$, distribution of random variables $\mathsf{Y}, \mathsf{M}_{INDEX}$ is independent of event $\boldsymbol{\rho} = i$.*

Proof. Conditioned on $\mathsf{T} = T$, string Y is chosen by picking an $n/2$ size set S uniformly at random from T, and setting these indices to 1, and this choice also fixes $\mathsf{M}_{\text{INDEX}}$ as the protocol is deterministic. And index ρ is chosen uniformly at random from T, independently of the choice of S, by definition of $\mathcal{D}'_\theta$. Thus, choice of $\boldsymbol{\rho} = i$ is independent of random variables $\mathsf{Y}, \mathsf{M}_{\text{INDEX}}$. ◀

Next, we will show that even conditioned on T, the entropy of Y remains large.

▷ Claim 18.

$$\mathbb{H}(\mathsf{Y} \mid \mathsf{T}) \geq \frac{n}{(1+2\theta)} \cdot H_2(1/2+\theta) - 2\log n.$$

Proof. We assume that $\theta < 1/2$, as otherwise, the statement is vacuously true. $H_2(1) = 0$ by definition, and entropy is always non-negative by Fact 7-(1).

Conditioned on $\mathsf{T} = T$, we know that Y is fixed by choosing set S uniformly at random. Thus,

$$\mathbb{H}(\mathsf{Y} \mid \mathsf{T}) = \log\left(\binom{b}{n/2}\right) \qquad \text{(by Fact 7-(1))}$$

$$\geq \log\left(2^{bH_2(n/2b)} \cdot \sqrt{\frac{b}{8(n/2)(b-n/2)}}\right) \quad \text{(by Fact 4, and } n/2 < b \text{, as } \theta < 1/2)$$
$$= b \cdot H_2(n/2b) + \frac{1}{2} \cdot \log(\frac{b}{8(n/2)(b-n/2)})$$
$$= \frac{n}{(1+2\theta)} \cdot H_2(1/2+\theta) + \frac{1}{2}\log(\frac{1}{4n \cdot (1/2-\theta)})$$
$$\geq \frac{n}{(1+2\theta)} \cdot H_2(1/2+\theta) + \frac{1}{2}\log(1/4n) \quad \text{(as } 1/2-\theta \leq 1)$$
$$\geq \frac{n}{(1+2\theta)} \cdot H_2(1/2+\theta) - 2\log n. \qquad \lhd$$

We are ready to prove Lemma 14.

Proof of Lemma 14. We can lower bound the entropy of W conditioned on $\mathsf{M}_{\text{INDEX}}, \boldsymbol{\rho}$ as,

$$\mathbb{H}(\mathsf{W} \mid \mathsf{M}_{\text{INDEX}}, \boldsymbol{\rho}) \geq \mathbb{H}(\mathsf{W} \mid \mathsf{M}_{\text{INDEX}}, \boldsymbol{\rho}, \mathsf{T}) \quad \text{(as conditioning reduces entropy, Fact 7-(2))}$$
$$= \mathop{\mathbb{E}}_{\mathsf{T}=T}\Big[\frac{1}{b} \cdot \sum_{\rho \in T} \mathbb{H}(\mathsf{W} \mid \mathsf{M}_{\text{INDEX}}, \boldsymbol{\rho} = \rho, \mathsf{T} = T)\Big] \quad \text{(as } \boldsymbol{\rho} \text{ is uniform over } T)$$
$$= \mathop{\mathbb{E}}_{\mathsf{T}=T}\Big[\frac{1}{b} \cdot \sum_{\rho \in T} \mathbb{H}(\mathsf{Y}(\rho) \mid \mathsf{M}_{\text{INDEX}}, \boldsymbol{\rho} = \rho, \mathsf{T} = T)\Big]$$
(as $W = Y(\rho)$ by definition of $\mathcal{D}_\theta$)
$$= \mathop{\mathbb{E}}_{\mathsf{T}=T}\Big[\frac{1}{b} \cdot \sum_{\rho \in T} \mathbb{H}(\mathsf{Y}(\rho) \mid \mathsf{M}_{\text{INDEX}}, \mathsf{T} = T)\Big]$$
(as $\mathsf{Y}, \mathsf{M}_{\text{INDEX}} \perp (\boldsymbol{\rho} = \rho) \mid \mathsf{T} = T$, by Observation 17)
$$\geq \mathop{\mathbb{E}}_{\mathsf{T}=T}\Big[\frac{1}{b} \cdot \mathbb{H}(\mathsf{Y}(T) \mid \mathsf{M}_{\text{INDEX}}, \mathsf{T} = T)\Big]$$
(by subadditivity of entropy, Fact 7-(4))
$$= \mathop{\mathbb{E}}_{\mathsf{T}=T}\Big[\frac{1}{b} \cdot \mathbb{H}(\mathsf{Y} \mid \mathsf{M}_{\text{INDEX}}, \mathsf{T} = T)\Big] \quad \text{(as } Y([n] \setminus T) \text{ is fixed to be 0)}$$
$$= \frac{1}{b} \cdot \mathbb{H}(\mathsf{Y} \mid \mathsf{M}_{\text{INDEX}}, \mathsf{T}).$$

Thus it follows that,

$$\mathbb{H}(\mathsf{Y} \mid \mathsf{M}_{\text{INDEX}}, \mathsf{T}) \leq b \cdot \mathbb{H}(\mathsf{W} \mid \mathsf{M}_{\text{INDEX}}, \boldsymbol{\rho}). \tag{1}$$

We also have,

$$\mathbb{H}(\mathsf{Y} \mid \mathsf{T}) = \mathbb{H}(\mathsf{Y}, \mathsf{M}_{\text{INDEX}} \mid \mathsf{T}) \quad \text{(as } M \text{ is fixed by } Y)$$
$$= \mathbb{H}(\mathsf{M}_{\text{INDEX}} \mid \mathsf{T}) + \mathbb{H}(\mathsf{Y} \mid \mathsf{M}_{\text{INDEX}}, \mathsf{T}) \quad \text{(by chain rule of entropy, Fact 7-(3))}$$
$$\leq \mathbb{H}(\mathsf{M}_{\text{INDEX}} \mid \mathsf{T}) + b \cdot \mathbb{H}(\mathsf{W} \mid \mathsf{M}_{\text{INDEX}}, \boldsymbol{\rho}) \quad \text{(by Eq (1))}$$
$$\leq \mathbb{H}(\mathsf{M}_{\text{INDEX}}) + b \cdot \mathbb{H}(\mathsf{W} \mid \mathsf{M}_{\text{INDEX}}, \boldsymbol{\rho}).$$
(as conditioning reduces entropy, by Fact 7-(2))

Combining with Claim 18, we get,

$$\mathbb{H}(\mathsf{M}_{\text{INDEX}}) + b \cdot \mathbb{H}(\mathsf{W} \mid \mathsf{M}_{\text{INDEX}}, \boldsymbol{\rho}) \geq b \cdot H_2(1/2+\theta) - 2\log n \quad \text{(as } b = n/(1+2\theta))$$
$$b \cdot \mathbb{H}(\mathsf{W} \mid \mathsf{M}_{\text{INDEX}}, \boldsymbol{\rho}) \geq b \cdot H_2(1/2+\theta) - (\mathbb{H}(\mathsf{M}_{\text{INDEX}}) + 2\log n).$$
(rearranging the terms)

Dividing both sides by b, we get,

$$\begin{aligned}\mathbb{H}(\mathsf{W} \mid \mathsf{M}_{\text{INDEX}}, \boldsymbol{\rho}) &\geq H_2(1/2+\theta) - \frac{1}{b} \cdot (\mathbb{H}(\mathsf{M}_{\text{INDEX}}) + 2\log n) \\ &\geq H_2(1/2+\theta) - \frac{2}{n} \cdot (\mathbb{H}(\mathsf{M}_{\text{INDEX}}) + 2\log n), \qquad (\text{as } b \geq n/2)\end{aligned}$$

finishing the proof. ◀

3.4 Extension to Augmented Chain

In this subsection, we will extend our lower bound to the Augmented Chain problem introduced by [15]. We begin by defining Augmented Index.

Augmented Index is a close variant of the INDEX problem. Here, in addition to having the index σ, Bob also has the bits $X(<\sigma)$. We know that this generalization also requires $\Omega(n)$ communication when Alice sends a message to Bob [34]. This problem is particularly useful for proving lower bounds for turnstile streams (see e.g., [11, 25, 16], and references therein). Tight information cost trade-off for this variant in the two-way communication model was proved by [9].

The formal definition of Augmented Chain follows.

▶ **Definition 19** (Augmented Chain). *The* AUG-CHAIN$_{n,k}$ *communication problem is defined as follows. Given $k+1$ players $\mathcal{P}_i$ for $i \in [k+1]$ where,*

- *$\mathcal{P}_i$ has a string $X_i \in \{0,1\}^n$ for each $i \in [k]$,*
- *$\mathcal{P}_i$ for $1 < i \leq k+1$ has an index $\sigma_{i-1} \in [n]$, and a string $X_{i-1}(<\sigma_{i-1})$,*

such that,

$$X_i(\sigma_i) = z,$$

for some bit $z \in \{0,1\}$. The players have a blackboard visible to all the parties. For $i \in [k]$ in ascending order, $\mathcal{P}_i$ sends a single message M_i, after which the index σ_i, and the string $X_{i-1}(<\sigma_{i-1})$ are revealed to the blackboard at no cost. $\mathcal{P}_{k+1}$ has to output whether z is 0 or 1. Refer to Figure 3 for an illustration.

For the chained version of Augmented Index, the lower bound that at least one player sends $\Omega(n/k^2)$ bits holds true, and the proof follows with minimal changes. Using this, [15] proved lower bounds for interval independent set selection in turnstile streams with weighted intervals.

We prove the following result about AUG-CHAIN$_{n,k}$.

▶ **Theorem 20.** *For any $n, k \geq 1$, any protocol for* AUG-CHAIN$_{n,k}$ *with probability of success at least 2/3 requires communication $\Omega(n - k\log n)$.*

A proof sketch detailing the changes needed to prove Theorem 20 is given in the full version. Most parts of the proof are similar to the proof of Theorem 3.

4 Applications to Streaming

In this section we give applications of our main result to independent sets in vertex arrival streams and streaming submodular maximization.

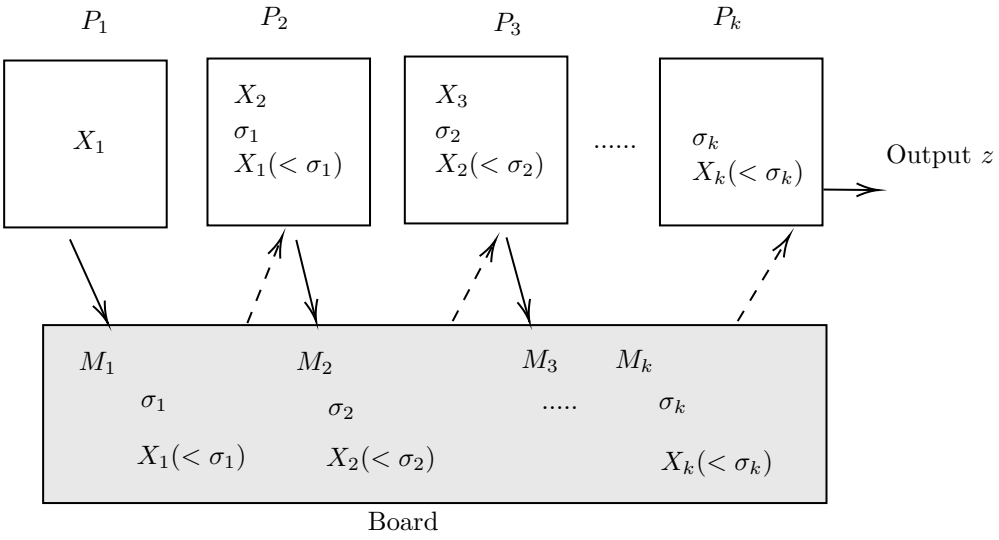


Figure 3 An illustration of the AUG-CHAIN$_{n,k}$ problem from Definition 19. The solid arrows illustrate that player $\mathcal{P}_i$ writes a message M_i to the board. The dashed arrows indicate that $\mathcal{P}_i$ can read the contents of the board. The order in which the messages are sent and indices, strings are released is also shown.

4.1 Independent Sets

In edge arrival streams, for any graph $G = (V, E)$, the vertex set V with n vertices is given, and the edges E arrive in any arbitrary order. We are required to process the graph in limited space.

In vertex arrival streams, for any graph $G = (V, E)$, the edges are grouped by their incident vertices. Vertices from V arrive one by one (in any arbitrary order), and when a vertex arrives, all the edges connecting it to any previously arrived vertices are revealed. This makes the vertex arrival stream a strictly easier model than the edge arrival stream, as the order of edges is restricted.

Indeed, for the maximal independent set problem, we know that finding algorithms in vertex arrival streams is easier; the greedy algorithm produces a maximal independent set in $\widetilde{O}(n)$ space, whereas, in edge arrival streams, any algorithm which finds a maximal independent set requires $\Omega(n^2)$ space [4, 12].

Maximum independent set (MIS), however, is provably hard in both vertex arrival streams and edge arrival streams. It is known that, any algorithm which performs an α-approximation of MIS in edge arrival streams requires $\Omega(n^2/\alpha^2)$ space from [22]. In vertex arrival streams, [12] proved a lower bound of $\Omega(n^2/\alpha^7)$. They also gave the following connection between CHAIN problem and MIS in the proof of Theorem 9 of their paper.

▶ **Proposition 21** (Rephrased from [12]). *For any $\alpha \geq 1$, any algorithm that gives an α-approximation of maximum independent sets in vertex arrival streams for n-vertex graphs using space at most s and probability of success at least 2/3 can be used to solve* CHAIN$_{n^2/64\alpha^4, 2\alpha}$ *with communication at most $2\alpha \cdot s$ bits and success probability at least 2/3.*

Our lower bound Theorem 3, along with Proposition 21 directly gives the following corollary.

▶ **Corollary 22.** *For $\alpha \geq 1$, any α-approximation of maximum independent sets in n-vertex graphs in vertex arrival streams uses $\Omega(n^2/\alpha^5 - \log n)$ space.*

This further reduces the gap between the lower bounds for α-approximation of MIS in vertex arrival streams and edge arrival streams by an α^2 factor.

4.2 Submodular Maximization

In this subsection, we will summarize our slight improvements to lower bounds for streaming submodular maximization.

A function f over ground set V from $f : 2^V \to \mathbb{R}$ is submodular if and only if, for any two sets $A \subset B \subset V$ and for any element $x \in V \setminus B$,

$$f(B \cup \{x\}) - f(B) \leq f(A \cup \{x\}) - f(A).$$

This captures the diminishing returns property of any submodular function.

We are interested in maximization of a monotone submodular function subject to a cardinality constraint. That is, for a given ℓ, we want to find a subset $S \subset V$ with $|S| \leq \ell$ such that for any other set $T \subset V$ with $|T| \leq \ell$, $f(S) \geq f(T)$.

We are given oracle access to function f, however, we do not have access to the entirety of the ground set. The elements of the ground set V arrive one by one, and the algorithm has space s to either store the incoming element or to discard it. The algorithm can query the oracle to f with any subset of the elements currently in storage. We want the storage to be roughly the same as the output size, which is ℓ.

In this model, [29] gave an algorithm which finds a $(1/2 - \epsilon)$-approximation in $O(\ell/\epsilon)$ space. [19] showed that a better approximation was not possible. They proved that any algorithm which gets a $(1/2 + \epsilon)$-approximation uses $\Omega(\epsilon |V| / \ell^3)$. They give the following connection to the CHAIN problem in Theorem 1.3 and Theorem 1.4 of their paper.

▶ **Proposition 23** (Rephrased from [19]). *For any $\epsilon > 0$, there exists a constant ℓ_0 such that for any $\ell \geq \ell_0$, any randomized streaming algorithm which maximizes a monotone submodular function $f : 2^V \to \mathbb{R}$ subject to cardinality constraint of at most ℓ, using space at most s and with approximation factor at least $(1/2 + \epsilon)$ in expectation can be used to solve $\text{CHAIN}_{|V|/\ell,\ell}$ with probability of success at least 2/3 and communication at most $s \cdot O(\ell/\epsilon)$.*

We get an improvement of ℓ factor over the current state-of-art lower bound in [19] as a corollary of Theorem 3.

▶ **Corollary 24.** *For any $\epsilon > 0$, there exists a constant ℓ_0 such that for any $\ell \geq \ell_0$, any streaming algorithm that maximizes a monotone submodular function $f : 2^V \to \mathbb{R}$ subject to a cardinality constraint of at most ℓ, with an approximation factor at least $(1/2 + \epsilon)$, requires $\Omega(|V| \epsilon/\ell^2 - \varepsilon \log(|V|))$ space.*

References

1. Farid Ablayev. Lower bounds for one-way probabilistic communication complexity and their application to space complexity. *Theoretical Computer Science*, 157(2):139–159, 1996. doi:10.1016/0304-3975(95)00157-3.
2. S. Assadi. Lecture notes on sublinear algorithms. https://sepehr.assadi.info/courses/cs514-s20/lec8.pdf, 2020.
3. S. Assadi, G. Kol, R. R. Saxena, and H. Yu. Multi-pass graph streaming lower bounds for cycle counting, max-cut, matching size, and other problems. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 354–364, Los Alamitos, CA, USA, November 2020. IEEE Computer Society. doi:10.1109/FOCS46700.2020.00041.

4 Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Sublinear algorithms for $(\Delta + 1)$ vertex coloring. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 767–786, 2019. doi:10.1137/1.9781611975482.48.

5 Sepehr Assadi, Sanjeev Khanna, and Yang Li. Tight bounds for single-pass streaming complexity of the set cover problem. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 698–711, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2897518.2897576.

6 Sepehr Assadi and Janani Sundaresan. (noisy) gap cycle counting strikes back: Random order streaming lower bounds for connected components and beyond. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pages 183–195, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3564246.3585192.

7 Sujoy Bhore, Fabian Klute, and Jelle J. Oostveen. On streaming algorithms for geometric independent set and clique. In Parinya Chalermsook and Bundit Laekhanukit, editors, *Approximation and Online Algorithms*, pages 211–224, Cham, 2022. Springer International Publishing. doi:10.1007/978-3-031-18367-6_11.

8 Harry Buhrman and Ronald Wolf. Communication complexity lower bounds by polynomials. In *Proceedings of the Annual IEEE Conference on Computational Complexity*, pages 120–130, February 2001. doi:10.1109/CCC.2001.933879.

9 Amit Chakrabarti, Graham Cormode, Ranganath Kondapally, and Andrew McGregor. Information cost tradeoffs for augmented index and streaming language recognition. *SIAM Journal on Computing*, 42(1):61–83, 2013. doi:10.1137/100816481.

10 Lijie Chen, Gillat Kol, Dmitry Paramonov, Raghuvansh R. Saxena, Zhao Song, and Huacheng Yu. Near-optimal two-pass streaming algorithm for sampling random walks over directed graphs. In *International Colloquium on Automata, Languages and Programming*, 2021. URL: https://api.semanticscholar.org/CorpusID:232014583.

11 Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, pages 205–214, New York, NY, USA, 2009. Association for Computing Machinery. doi:10.1145/1536414.1536445.

12 Graham Cormode, Jacques Dark, and Christian Konrad. Independent Sets in Vertex-Arrival Streams. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 45:1–45:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2019.45.

13 Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

14 Carsten Damm, Stasys Jukna, and Jirí Sgall. Some bounds on multiparty communication complexity of pointer jumping. In Claude Puech and Rüdiger Reischuk, editors, *STACS 96, 13th Annual Symposium on Theoretical Aspects of Computer Science, Grenoble, France, February 22-24, 1996, Proceedings*, volume 1046 of *Lecture Notes in Computer Science*, pages 643–654. Springer, 1996. doi:10.1007/3-540-60922-9_52.

15 Jacques Dark, Adithya Diddapur, and Christian Konrad. Interval Selection in Data Streams: Weighted Intervals and the Insertion-Deletion Setting. In Patricia Bouyer and Srikanth Srinivasan, editors, *43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2023)*, volume 284 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:17, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.FSTTCS.2023.24.

16 Jacques Dark and Christian Konrad. Optimal lower bounds for matching and vertex cover in dynamic graph streams. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 30:1–30:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.CCC.2020.30.

17 Ashkan Norouzi Fard, Moran Feldman, Ola Svensson, and Rico Zenklusen. Submodular maximization subject to matroid intersection on the fly. In *30th Annual European Symposium on Algorithms, ESA 2022, September 5-9, 2022, Berlin/Potsdam, Germany*, pages 52:1–52:14, 2022. doi:10.4230/LIPICS.ESA.2022.52.

18 Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. Graph distances in the data-stream model. *SIAM J. Comput.*, 38(5):1709–1727, 2008. doi:10.1137/070683155.

19 Moran Feldman, Ashkan Norouzi-Fard, Ola Svensson, and Rico Zenklusen. The one-way communication complexity of submodular maximization with applications to streaming and robustness. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pages 1363–1374, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3357713.3384286.

20 Sudipto Guha and Andrew McGregor. Stream order and order statistics: Quantile estimation in random-order streams. *SIAM J. Comput.*, 38(5):2044–2059, 2009. doi:10.1137/07069328X.

21 Venkatesan Guruswami and Krzysztof Onak. Superlinear lower bounds for multipass graph processing. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, pages 287–298, 2013. doi:10.1109/CCC.2013.37.

22 Magnús M. Halldórsson, Xiaoming Sun, Mario Szegedy, and Chengu Wang. Streaming and communication complexity of clique approximation. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 449–460. Springer, 2012. doi:10.1007/978-3-642-31594-7_38.

23 P. Indyk and D. Woodruff. Tight lower bounds for the distinct elements problem. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 283–288, 2003. doi:10.1109/SFCS.2003.1238202.

24 Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A property of quantum relative entropy with an application to privacy in quantum communication. *J. ACM*, 56(6), September 2009. doi:10.1145/1568318.1568323.

25 Daniel M. Kane, Jelani Nelson, and David P. Woodruff. On the exact space complexity of sketching and streaming small norms. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 1161–1178, USA, 2010. Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611973075.93.

26 Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Streaming lower bounds for approximating MAX-CUT. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1263–1282. SIAM, 2015. doi:10.1137/1.9781611973730.84.

27 Michael Kapralov, Amulya Musipatla, Jakab Tardos, David P. Woodruff, and Samson Zhou. Noisy Boolean Hidden Matching with Applications. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 91:1–91:19, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2022.91.

28 Mikhail Kapralov. Better bounds for matchings in the streaming model. In *ACM-SIAM Symposium on Discrete Algorithms*, 2012. URL: https://api.semanticscholar.org/CorpusID:448251.

29 Ehsan Kazemi, Marko Mitrovic, Morteza Zadimoghaddam, Silvio Lattanzi, and Amin Karbasi. Submodular streaming in all its glory: Tight approximation, minimum memory and low adaptive complexity. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3311–3320. PMLR, 09–15 June 2019. URL: https://proceedings.mlr.press/v97/kazemi19a.html.

30 Ilan Kremer, Noam Nisan, and Dana Ron. On randomized one-round communication complexity. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '95, pages 596–605, New York, NY, USA, 1995. Association for Computing Machinery. doi:10.1145/225058.225277.

31 Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

32 F. J. (Florence Jessie) MacWilliams and N. J. A. (Neil James Alexander) Sloane. *The theory of error-correcting codes*. North-Holland mathematical library ; v. 16. North-Holland Pub. Co., Amsterdam, 1978 - 1977.

33 Guangxu Yang Mi-Ying Huang, Xinyu Mao and Jiapeng Zhang. Breaking square-root loss barriers via min-entropy. *Electron. Colloquium Comput. Complex.*, pages TR24–067, 2024. URL: https://eccc.weizmann.ac.il/report/2024/067/.

34 Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. *Journal of Computer and System Sciences*, 57(1):37–49, 1998. doi:10.1006/jcss.1998.1577.

35 Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020. doi:10.1017/9781108671644.

36 Mert Saglam. Tight bounds for data stream algorithms and communication problems. Master's thesis, Simon Fraser University, 2011.

Polynomial Size, Short-Circuit Resilient Circuits for NC

Yael Tauman Kalai
Microsoft Research, Cambridge, MA, USA
MIT, Cambridge, MA, USA

Raghuvansh R. Saxena
Tata Institute of Fundamental Research, Mumbai, India

Abstract

We show how to convert any circuit of poly-logarithmic depth and polynomial size into a functionally equivalent circuit of polynomial size (and polynomial depth) that is resilient to adversarial short-circuit errors. Specifically, the resulting circuit computes the same function even if up to ϵd gates on every root-to-leaf path are short-circuited, *i.e.*, their output is replaced with the value of one of its inputs, where d is the depth of the circuit and $\epsilon > 0$ is a fixed constant.

Previously, such a result was known for formulas (Kalai-Lewko-Rao, FOCS 2012). It was also known how to convert general circuits to error resilient ones whose size is quasi-polynomial in the size of the original circuit (Efremenko et al. STOC 2022). The reason both these works do not extend to our setting is that there may be many paths from the root to a given gate, and the resilient circuits needs to "remember" a lot of information about these paths, which causes it to be large. Our main idea is to reduce the amount of this information at the cost of increasing the depth of the resilient circuit.

2012 ACM Subject Classification Theory of computation → Communication complexity; Theory of computation → Circuit complexity

Keywords and phrases Error-resilient computation, short-circuit errors

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.90

Funding *Raghuvansh R. Saxena*: Supported by the Department of Atomic Energy, Government of India, under project no. RTI4001.

1 Introduction

Constructing error-resilient circuits is one of the most fundamental problems in theoretical computer science, dating back to von Neumann [30]. In the study of error resilient circuits, the goal is to convert any circuit C into a circuit C' that computes the same function as C even if some of the gates of C' are faulty. Moreover, the goal is to do so with a small overhead in size, *i.e.*, constructing a circuit C' whose size is bounded by a polynomial in the size of C.

In this paper, we show how to convert any polynomial size circuit of poly-logarithmic depth into a polynomial size circuit that is functionally equivalent and is resilient to short-circuit errors, an error model that was introduced by Kleitman, Leighton, and Ma [20], and was soon adopted as a central model in the study of error-resilient circuits [18, 4, 7]. In this error model, the adversary can replace any gate in the circuit with an arbitrary gate $g : \{0,1\}^2 \to \{0,1\}$ as long as $g(0,0) = 0$ and $g(1,1) = 1$ (and in particular can replace an AND gate with an OR gate, and vice versa)[1]. Equivalently, we can say that the value of the gate g is replaced by the value of one of its children (the wire to the other child is "cut

[1] Note that the restriction that $g(b,b) = b$ for all $b \in \{0,1\}$ is necessary since otherwise an adversary can simply flip the result of the output gate, and thus no circuit can be resilient to even a single error.


© Yael Tauman Kalai and Raghuvansh R. Saxena;
licensed under Creative Commons License CC-BY 4.0



out"). This model is motivated by applications; indeed, as noted in [20], "stuck-at" and "power-ground" failures resulting from short-circuits or broken connections are more common than other types of errors.

Similar to the works of [18, 4, 7], we consider omniscient adversaries that have full information about the entire circuit, and can corrupt ϵd of the gates on every root-to-leaf path, where d is the depth of the circuit and $\epsilon > 0$ is some fixed constant. These prior works were either restricted to formulas, *i.e.*, logarithmic depth circuits [18, 4], or incurred a quasi-polynomial blowup to the circuit size [7]. Moreover, it was conjectured in [7] that, unlike the case of formulas [18, 4], a quasi-polynomial blowup in the circuit size is necessary when converting general circuits into noise resilient ones.

1.1 Our Results

We show that how to convert any circuit of polynomial size and poly-logarithmic depth (i.e., any NC circuit) into a functionally equivalent circuit of polynomial size (and polynomial depth) that is error resilient.

▶ **Theorem 1** (Informal, see formal version in Theorem 8)**.** *Let C be a Boolean circuit in the class NC that[2] computes a function f. There exists a Boolean circuit C' whose size and depth are polynomial in the size of C that computes f even when a constant fraction of gates in any input to output path in C' are short-circuited.*

In terms of comparison, Theorem 1 improves on the main result of [18] in the sense that it works for all circuits with polynomial size and poly-logarithmic depth while [18] only works for formulas. It also improves upon [7] in the sense that it gets a polynomial blowup in the size instead of the quasi-polynomial blowup obtained in [7]. However, it does not subsume either of these results. For example, [18] also preserve the depth of the original formula up to constant factors but we do not provide such a guarantee, and [7] works for general circuits, including those not in the class NC.

We also mention that [7] believed that the quasi-polynomial blowup they obtain is inherent and conjectured that a polynomial blowup will not suffice like it does in the case of formulas [18, 4]. Theorem 1 disproves this conjecture for the restricted case of circuits in NC. Proving/disproving the conjecture for general circuits is an amazing problem that we leave open. However, proving this conjecture, even in an existential manner, would imply $\mathsf{P/poly} \not\subseteq \mathsf{NC}$, and thus may be extremely hard. Indeed, if $\mathsf{P/poly} \subseteq \mathsf{NC}$, then a general polynomial size circuit has a functionally equivalent circuit in NC and we can use Theorem 1 to make it error resilient with only a polynomial blowup.

Error-resilient dag-protocols

Computation and communication have a deep connection. Just like prior work [18, 4, 7], the current work also exploits this connection via the Karchmer-Wigderson transformation. Specifically, Karchmer and Wigderson [19] show that any circuit computing a function f can be converted into a communication protocol for a related search problem KW_f such that the depth of the circuit matches the length of the communication protocol, and vice versa.

[2] Thus, if f takes n variables as input, the number of gates in C is bounded by a polynomial in n and the depth of C is bounded by a polynomial in $\log n$. The other constants in the theorem statement depend on the degree of these polynomials; see the formal version of the theorem (Theorem 8) for the precise dependence.

An analogous transformation can be shown in the reverse direction, *i.e.* a transformation that takes error-resilient communication protocols to error resilient circuits, and was used in [18, 4] to build error resilient formulas.

The Karchmer-Wigderson transformation does not preserve the size of the circuit and in fact, blows it up to be exponential in the depth. To circumvent this blowup in size, [25, 29] showed how to convert any circuit into a "**dag**-protocol" that preserves both its size and depth. **dag**-protocols are a generalization of communication protocols, and can be represented by the following pebble game: There is a rooted directed acyclic graph, and each non-sink node in the graph belongs to one of the two parties, while each sink node is associated with an output. The game starts with a "pebble" at the root of graph, that is moved along the edges of the graph. At every step, if the pebble is not already at a sink node, the party who owns that node will move the pebble along one of its out-edges. This will end when the pebble reaches a sink node, and the output will be the output of the sink node.

[25, 29] showed that every circuit computing a function f can be converted to a **dag**-protocol with the same size and depth that solves KW_f with a strong correctness guarantee called rectangular correctness (see Definition 5 for a precise definition). The reverse direction also holds, and in fact even holds in the error-resilient setting. This direction was used to get error-resilient circuits from error-resilient **dag**-protocols in [7]. We adopt a similar approach and use the [25, 29] transformation to go from circuits to **dag**-protocols and back. Indeed, our main technical result is the following theorem about error-resilient **dag**-protocols.

▶ **Theorem 2** (Informal, see formal version in Theorem 9)**.** *Let S be a search problem and Π be a* **dag***-protocol that solves S with rectangular correctness. Assume that the depth of Π is poly-logarithmic in the size. There exists a* **dag***-protocol Π' with a polynomially larger size that solves S with rectangular correctness even if a constant fraction of nodes on any root to leaf path in the protocol Π' are adversarially corrupted.*

1.2 Related Work

The works most closely related to ours are the works [18, 4, 7] cited above. We discuss additional related work below.

Other noise models for circuits

Even though the short-circuit error model we consider is the most studied one, other models have also been considered. For example, von Neumann [30] initiated the study of the stochastic noise model, where the noise flips the value of each gate in the circuit independently with some small fixed probability. Von Neumann's model was studied by a long sequence of works, including [6, 22, 23, 11, 8, 9, 17, 13, 12, 10]. In this model, it is known that a circuit of size s can converted to a noise resilient circuit of size $O(s \log s)$, and that a function with sensitivity s' requires a resilient circuits of size $\Omega(s' \log s')$ [30, 6, 22, 13, 12].

In this paper, we study the short-circuit error model of [20] but there is also a different adversarial model studied by [14], where the adversary may corrupt the output of a small constant fraction of the gates at each layer of the circuit in an arbitrary way. They show how to construct error resilient circuits for symmetric functions in this model, by exploiting interesting connections between their model and probabilistically checkable proofs. However, the obtained circuit is only guaranteed to compute, what they call, a "loose version" of

the function, and may err on many inputs. Lastly, we mention the orthogonal direction of constructing testable circuits which are circuits on which errors can detected (but not necessarily corrected) efficiently [2, 1].

Codes for interactive communication

As our work develops error-resilient circuits via error-resilient **dag**-protocols, it is also connected with the vast literature on codes for interactive communication, or simply "interactive codes", that are used to make communication protocols resilient to noise. This field, initiated by the seminal works [26, 27, 28] has received a lot of attention over the last three decades, and various aspects of interactive codes have been extensively studied. A work loosely related to circuits is [5] but see [16] for an excellent survey. We note that all constructions of error resilient **dag**-protocols are heavily inspired by an underlying interactive coding scheme (recall that **dag**-protocols are a generalization of communication protocols).

dag-protocols

Razborov [25] introduced a model of *PLS communication protocols*, and used it generalize the Karchmer-Wigderson transformation [19] in a way that preserves both the size and the depth of the circuit. This connection was used by Krajícek [21], who introduced the technique of *monotone feasible interpolation*, which became a popular method for proving lower bounds on the refutation size in propositional proof systems such as Resolution, and Cutting Planes [3], by reducing to monotone circuit lower bounds. The notion of PLS communication protocols was simplified by Pudlak [24] and Sokolov [29] to the notion of **dag**-*like communication protocols*, which we call **dag**-protocols. Subsequently, a "converse" to monotone feasible interpolation was established in [15] to prove new lower bounds on monotone circuits by lifting lower bounds on Resolution refutations. To the best of our knowledge, our work is the only work, other than [7], to use the notion of **dag**-protocols in a "positive" manner, namely to construct error-resilient circuits, in contrast to prior work which mainly used this notion to prove lower bounds in proof complexity and circuit complexity.

1.3 Additional Discussion

An obvious problem left open by our work is whether general circuits can be made error resilient with only a polynomial overhead. As mentioned above, the only work in this regard is [7], that showed that general circuits can be made error-resilient by incurring a quasi-polynomial overhead. Specifically, if the original circuit has size s and depth d, they construct a resilient version of this circuit that has size $s^{\mathcal{O}(\log d)}$ and depth $\mathcal{O}(d)$. [7] further say that the power of $\mathcal{O}(\log d)$ seems inherent.

The current work manages to get around the $\mathcal{O}(\log d)$ in the exponent, at the cost of increasing depth to be a polynomial (note that $d = (\log s)^{\mathcal{O}(1)}$ in our setting). It is interesting to see if our techniques can be combined with [7] to shave off a small factor off of this exponent for general circuits, at the expense of a larger depth. We leave this investigation to future work. Another great question for future research is whether the polynomial blowup in depth that we suffer is inherent, or is there a way to make circuits in NC error resilient while preserving their depth up to a constant factor.

Another interesting direction for future work is whether the resilient circuit can be computed efficiently, say in time polynomial in the size of the circuit. The initial work of [18] guaranteed efficiency when the initial circuit is a formula although the later work [7]

was not able to generalize it to circuits[3]. The current work for circuits in NC also leaves this direction open. Finally, even though we get resilience to some constant fraction of adversarial errors, we make no efforts to optimize this constant. Finding the right constant is a great question for future work (see [4]).

2 Technical Overview

We now overview the ideas behind Theorem 1 and contrast it with prior work [18, 4, 7]. Roughly speaking, Theorem 1 says that there exist polynomial sized resilient circuits for every circuit C in NC. We let s and d be the size and the depth of C, respectively, and use f to denote the Boolean function computed by C. We will also assume without loss of generality that the circuit C is layered and alternating. Just like prior work, our construction is in three-steps via **dag**-protocols[4]:

1. **Circuits → dag-protocols.** This step is the same as prior work, and uses the Karchmer-Wigderson transformation [19] to transform the circuit that computes f to a **dag**-protocol that solves the Karchmer-Wigderson game KW_f associated with f with rectangular correctness. This transformation satisfies the property that short-circuit errors in the original circuit correspond to running the protocol over a channel with corruption noise and perfect feedback. We will call such channels feedback channels for simplicity and use this property in the Step 3 below.
2. **dag-protocols → error-resilient dag-protocols.** This is the main step in the proof where the input is a **dag**-protocol and the goal is to output a functionally equivalent **dag**-protocol that works on feedback channels. This is the main contribution of this work and is detailed below.
3. **Error-resilient dag-protocols → error-resilient circuits.** Having constructed a protocol that solves KW_f with rectangular correctness even on feedback channels, we use the "inverse" Karchmer-Wigderson transformation to transform it to a Boolean circuit computing f that is resilient to short-circuit errors. This step in our proof can also be found in prior work.

The main contribution of this work is Step 2 above, which transforms **dag**-protocols to error-resilient **dag**-protocols. Analogous transformations in prior work were usually done using interactive coding schemes, an area of research initiated by the seminal works [26, 27, 28] that is dedicated to making communication protocols resilient to errors. At an extremely high level, this is done by designing mechanisms to detect errors fast and then using rewind mechanisms to rewind to a point in the communication history that was before these errors occurred.

However, to rewind to a point in the communication history before the errors occurred, the resilient protocol needs to remember the state at that point. After the Karchmer-Wigderson transformation, this extra memory shows up as a blowup in the size of the resilient circuit, which we want to minimize. To be more specific, prior work on resilient circuits [7], remembers $\mathcal{O}(\log d)$ different states at every point in the resilient protocol which means that each layer in the resilient circuit is quasi-polynomially larger (*i.e.*, has size $s^{\mathcal{O}(\log d)}$) than each layer in the original circuit. Thus, even though [7] increase the number of layers by only a constant factor, the blowup in the size of each layer makes the overall blowup quasi-polynomial.

[3] Note that the error-resilient circuit of [7] is of quasi-polynomial size, and thus polynomial in this size of this circuit means that it is quasi-polynomial in the size of the original circuit.
[4] Readers not familiar with **dag**-protocols may find it useful to first go over Section 2.1 of [7].

Our approach is to greatly reduce the size of every layer, at the expense of the increasing the total number of layers. This means that we cannot even afford to remember $\mathcal{O}(\log d)$ many states at any point. As any interactive coding scheme we are aware of remembers at least logarithmically many states, they would not suit us, and new ideas are needed.

2.1 Our Approach For NC^1

We start by considering a restricted case where the original circuit C lies in NC^1. Even though circuits in NC^1 can be transformed into formulas with only a polynomial blowup, we will stick with the circuit view as it will allow us to generalize later on. Note that prior work already gives a resilient circuit for this case that has size polynomial in s and has depth $\mathcal{O}(d)$. We construct a different resilient circuit that has the same size guarantee but relaxes the depth guarantee[5]. This alternate construction allows us to extend to all of NC.

Our approach

We devise a way to make **dag**-protocols corresponding to NC^1 circuits resilient while remembering only a constant number (in fact, 2) states at each point, at the cost of blowing up the number of layers exponentially. As the depth of NC^1 circuits is logarithmic in n, this exponential blowup is still polynomial in n. Moreover, as we only remember a constant number of states at each point, the blowup in the size of each layer is also polynomial, and combining this with the blowup in the number of layers still gives a polynomial blowup.

Note that any protocol must at least remember the current state, and thus a protocol that remembers only 2 states actually remembers only 1 extra state! Interestingly, for our protocol, this extra state is not a state in the past but actually the next state in the future that the protocol is considering going to. Specifically, if the protocol is currently in state in layer $i-1$ (for some $i > 0$), and wants to advance to a state in layer i, then instead of jumping to that state directly, it is remembered as a "candidate" state until the protocol has enough "confidence" in the candidate to advance. This confidence is implemented via a variable `cnf` that is incremented every time the parties want to advance and decremented every time they suspect that the states in memory are incorrect. Only after the confidence `cnf` hits a certain threshold $\mathfrak{C}_i$ does the protocol actually advance to the state in layer i.

We will view the thresholds as being cumulative, *i.e.*, $\mathfrak{C}_i$ is the total confidence required to go from layer 0 to layer i. This means that $\mathfrak{C}_i - \mathfrak{C}_{i-1}$ is the additional confidence needed to advance from layer $i-1$ to layer i. With this said, the obvious question is how large does $\mathfrak{C}_i - \mathfrak{C}_{i-1}$ need to be? In other words, how much confidence should the protocol have in the candidate before they advance to a state in layer i? To answer this question, one needs to consider the total "loss" of the protocol in case it advances wrongly and reaches an incorrect state. As no state in the past is remembered, the only way the protocol can recover from this incorrect state is if it starts all over again and reaches the state in layer $i-1$, which requires a total increase in confidence of $\mathfrak{C}_{i-1}$. Thus, the increase (which equals $\mathfrak{C}_i - \mathfrak{C}_{i-1}$) of the confidence required to jump from layer $i-1$ to layer i should be the same order of magnitude as the confidence $\mathfrak{C}_{i-1}$ needed to reach layer $i-1$ from the beginning of the protocol. Solving, we get that $\mathfrak{C}_i$ should be (at least) exponential in i. This is the choice that we make, setting $\mathfrak{C}_i = c^i$ for some suitable constant $c > 1$.

[5] The depth is at most the size, so it must still be polynomial in s.

Protocol details

We now provide more details of our protocol and argue its correctness. The protocol starts at the initial (root) node in layer 0 with confidence $\texttt{cnf} = \mathfrak{C}_0 = 0$. At any given time, the protocol is in some layer, say layer $i-1$ with some confidence $\texttt{cnf}$, and trying to continue the simulation from some state in this layer. For this, the protocol first checks[6] if the current state has any errors. If so, the protocol decreases the confidence $\texttt{cnf}$ by 1. This may cause $\texttt{cnf}$ to hit 0, which is taken as a signal to forget all progress and restart the simulation. If no errors are found, then the current state is assumed to be correct and the action to be taken is determined by the value of the confidence $\texttt{cnf}$.

If we have $\texttt{cnf} < \mathfrak{C}_{i-1}$, then it must be the case that the confidence was decreased after reaching layer $i-1$. The protocol simply increases it by 1 trying to get it back to $\texttt{cnf}_{i-1}$ so that further progress can be made. If the confidence $\texttt{cnf} = \mathfrak{C}_{i-1}$, the confidence in layer $i-1$ is high enough for the protocol to compute a candidate in layer i. A candidate $\texttt{nxt}$ for the state to advance to is computed and the confidence is further increased by 1.

Finally, if the confidence $\texttt{cnf} > \mathfrak{C}_{i-1}$, then the protocol must already have a candidate $\texttt{nxt}$ that it is considering going to. The protocol then checks if this candidate was caused due to errors, and if so, decreases the confidence by 1. If the confidence is lowered all the way back to $\mathfrak{C}_{i-1}$, the protocol "forgets" about the candidate $\texttt{nxt}$ (and will compute a new one in the future if needed). On the other hand, if the candidate $\texttt{nxt}$ was not due to errors, the protocol increases the confidence $\texttt{cnf}$ by 1. If this increased confidence reaches $\texttt{cnf}_i$, it is high enough to advance. The protocol forgets about the current state in layer $i-1$ and continue the rest of the simulation from the state $\texttt{nxt}$ in layer i.

Arguing correctness

We now show that the above protocol indeed simulates the original protocol correctly. For this, we first show that if the protocol has high confidence at the end of the protocol, specifically, if $\texttt{cnf} \geq \mathfrak{C}_d$, where d is the total number of layers (depth) of the original protocol, then the output of the protocol must be correct. Indeed, if the confidence is that high, the protocol is in a state in layer d or higher[7], and the only reason this state would not be correct is if the protocol advanced to it incorrectly. However, this means that there must have been at least $\mathfrak{C}_d - \mathfrak{C}_{d-1}$ errors while this state was the next candidate. As the confidence thresholds are exponential, we have that the total number of errors is at least $\mathfrak{C}_d - \mathfrak{C}_{d-1} = \Omega(\mathfrak{C}_d)$, which is more than a constant fraction and therefore, unaffordable.

Now, all we need to show is that the confidence is indeed high at the end of the protocol. For this, we fix any execution of the protocol and partition the rounds into rounds where the protocol wants to "go back" and decrease the confidence $\texttt{cnf}$ (this happens when either the current state or the next state $\texttt{nxt}$ has any errors) and rounds where it does not. We collect the former type of rounds in the set Back. By definition, the confidence will increase in any round not in Back unless there are errors, and will decrease in any round in Back

[6] As in prior work, assuming that the original **dag**-protocol is rectangular correct is crucial for the protocol to be able to check whether or not there are errors. As this is not a novel idea for the current work, we omit a precise description in this sketch. (Very) roughly speaking (see formal definition in Definition 5), a state is said to be rectangular correct for inputs x and y for Alice and Bob respectively, if there exists inputs x' for Alice and y' for Bob such that the protocol reaches that state when executed with inputs x and y' and also reaches it with inputs x' and y. This means that the parties can go over all possible inputs of the other party in order to check if the state is (rectangular) correct or not.

[7] We append the original protocol with dummy states to get states beyond layer d.

unless there are errors. As the total number of errors is only a small constant fraction, this means that all we need to do to show that the confidence is high at the end of the protocol is to show that the size of Back is small.

To show this, we argue that the size of Back is at most a constant factor more than the number of rounds in Back where the protocol did not go back. To see why, note that the first round in Back must be a round where the protocol added an incorrect next state as their candidate `nxt`. As mentioned above, if the next state is in layer i, this only happens when the confidence is exactly $\mathfrak{C}_{i-1}$.

As long as the protocol does not advance to this next state (in other words, it remains a candidate), the confidence must stay above $\mathfrak{C}_{i-1}$, as otherwise the protocol will forget the incorrect candidate and look for a new one. This means that the total number of rounds where the protocol does not go back and increase the confidence exceeds the total number of rounds where the protocol did go back, and we are done. On the other hand, if the protocol does advance to the candidate, there must have been at least $\mathfrak{C}_i - \mathfrak{C}_{i-1}$ errors, and by definition of $\mathfrak{C}$, these are high enough (up to constant factors) for $\mathfrak{C}_i$ rounds of progress. Thus, the protocol can wait until the confidence hits 0 and restart the simulation from there.

2.2 From NC^1 to NC

We now show how we can extend the ideas above to get a simulation for all of NC. Note that the Karchmer-Wigderson transformation transforms general NC circuits to **dag**-protocols of poly-logarithmic depth. As our arguments above increase the depth exponentially, using them on general NC circuits would make the depth super-polynomial.

To reduce the depth, we extend our protocol above to also remember some states in the past. Specifically, if the current state is layer $i-1$, the protocol also remembers the state it encountered in the most recent layer that was a multiple of $(\log s)^j$, for all $j \in \mathbb{N}$. As the total number of layers we have is poly-logarithmic, we get that when j is larger than some fixed constant, the most recent layer that is a multiple of $(\log s)^j$ is just layer 0. This means that the parties only remember a constant number of distinct states, implying as above that the blowup in the size of each layer is still polynomial.

We now argue why remembering these extra states ("meeting points") in the past will also reduce the total number of layers back to polynomial. The reason is that if the protocol remembers meeting point in the near future, then, when the confidence is too low, the protocol does not have to forget all progress and restart the simulation from scratch. Instead, the protocol only has to forget the progress since the last meeting point and restart the simulation from said meeting point. Because of the loss in progress is now smaller, we can allow the confidence increase (which equals $\mathfrak{C}_i - \mathfrak{C}_{i-1}$) required to advance to also be smaller. As this happens for almost all i, the value of $\mathfrak{C}_d$, the highest confidence threshold is significantly smaller, and we can bound it by a polynomial.

The new confidence thresholds

We are yet to specify the precise increase in confidence (which equals $\mathfrak{C}_i - \mathfrak{C}_{i-1}$) required to advance to a state in layer i. For this, we look at the representation of i as an integer[8] in base $\log s$. If $a_1, a_2, \ldots$ are the digits in this representation, we set $\mathfrak{C}_i - \mathfrak{C}_{i-1}$ to be proportional to $c^{\sum_i a_i}$. As there is a meeting point in memory for every power of $\log s$, it can be seen

[8] The actual proof uses the representation of $i-1$ for technical reasons.

that $c^{\sum_i a_i}$ takes into account the distance from all the meeting points stored in memory, while maintaining the exponential growth required between two meeting points. We defer the remaining details to the proof.

Getting rectangular correctness

Just like prior work, for Step 2 above to work, it is crucial that our simulation protocol is rectangular correct, which is a stronger notion of correctness than standard correctness, and is necessary for Step 3 to work. Without going into the precise definition, this means that we need to account for the errors in the rounds where the parties (Alice and Bob) are talking separately. That is, the errors in rounds where Alice is talking cannot be used to offset the rounds where Bob is going back and decreasing the confidence, and vice versa.

Other than doubling the notation we use (*e.g.*, we now need to remember a pair of meeting points, one for Alice and one for Bob, *etc.*), this also introduces a more subtle challenge regarding the fraction of errors that we can be resilient to. To understand this, recall that the layers in the original circuit, and therefore the layers in the original **dag**-protocol, are alternating. This means that the party controlling the layer changes every time the protocol advances a layer. Now, imagine what happens when the adversary spends $\mathfrak{C}_i - \mathfrak{C}_{i-1}$ errors in a layer controlled by Alice to advance it incorrectly to a layer controlled by Bob.

As the errors for the both the parties are accounted for separately, Bob cannot detect and correct these errors, and the protocol will run as if there were no errors till the next time Alice gets to speak. This happens only after $\mathfrak{C}_{i+1} - \mathfrak{C}_i$ rounds, which due to our definition of $\mathfrak{C}$, may be a factor of c larger than $\mathfrak{C}_i - \mathfrak{C}_{i-1}$, the total number of errors inserted by the adversary. Thus, it is possible for the adversary to waste c rounds of simulation per error, implying that there has to be a multiplicative $1/c$ loss in the overall error resilience. Moreover, c cannot be too small, as then the parties will not be able to correctly maintain their meeting points. We are able to show that this loss is a constant proportional to the total number of meeting points; see the formal version of our main theorem (Theorem 8).

3 Model and Preliminaries

We now describe the model and our result more formally. Note that our description closely follows [7].

3.1 Circuits and dag-protocols

Circuits

We will consider Boolean circuits with negations at the inputs. That is, a Boolean circuit will be a directed acyclic graph C where each non-source node (or gate) is labelled as an $\wedge$ gate or an $\vee$ gate, while each source node is labelled with an input variable or the negation of the input variable. The source nodes are also called input gates and one designated node in C is called the output gate. We use $V_\wedge$ and $V_\vee$ to denote the set of all the nodes in C that are labelled $\wedge$ and $\vee$ respectively. We define how computation over a circuit C takes place by inductively defining the functions $f_{C,v}$ computed at each gate v. We have:

$$f_{C,v} = \begin{cases} z, & \text{if } v \text{ is an input gate with label } z \\ \bigvee_{u:(u,v) \text{ is an edge in } C} f_{C,u}, & \text{if } v \in V_\vee \\ \bigwedge_{u:(u,v) \text{ is an edge in } C} f_{C,u}, & \text{if } v \in V_\wedge \end{cases} . \tag{1}$$

When v is the output gate, we omit it from the notation and simply write f_C, which we call the function computed by C. We will assume that every gate in C that is not an input gate has at least one edge coming to it. We define the size of C to be the number of nodes in C and the depth of C to be the length of the longest path in C starting from an input gate and ending at the output gate. We use $\|C\|$ to denote the size of C.

Note that edges in C are going "up" from the inputs to the output gate. This contrasts with the edges in **dag**-protocols that are going "down" from the root to the leaves as defined below. These two directions, although different, follow the conventions for both circuits and **dag**-protocols.

Search problems and KW-games

Let $\mathcal{X}$ and $\mathcal{Y}$ be input sets for Alice and Bob respectively and $\mathcal{O}$ be a set of outputs. A search problem on these input and output sets is defined by a relation $S \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{O}$, and the goal of Alice and Bob is to determine, given inputs $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ respectively, an element $o \in \mathcal{O}$ such that $(x, y, o) \in S$.

KW-games are special type of search problems that are defined using a Boolean function f on n variables (for some n). Here, the set $\mathcal{X}$ is the set $f^{-1}(1)$ of all inputs for which f evaluates to 1, $\mathcal{Y}$ is the set $f^{-1}(0)$ of all inputs for which f evaluates to 0, and $\mathcal{O}$ is the set $[n]$. As $\mathcal{X}$ and $\mathcal{Y}$ are disjoint by definition, for any $x \in \mathcal{X}$, $y \in \mathcal{Y}$, there exists $o \in [n]$ such that $x_o \neq y_o$. The search problem KW_f is the problem of finding such an o, namely, it is the subset:

$$\mathsf{KW}_f = \{(x, y, o) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{O} \mid x_o \neq y_o\}.$$

dag-protocols

We now define the notion of **dag**-protocols. Let $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{O}$ be input and output sets as above. A **dag**-protocol is defined by a tuple:

$$\Pi = \left(G = (V_A \cup V_B \cup V_O, E), \mathsf{rt}, \{h_v\}_{v \in V_A \cup V_B}, \{o_v\}_{v \in V_O}\right).$$

Here, G is a directed acyclic graph with vertices partitioned into V_A, V_B, and V_O respectively, and edges E. We will use $V = V_A \cup V_B \cup V_O$ to be the set of all vertices in G and $\mathsf{rt} \in V$ is a special vertex called the root. For all $v \in V_A$, the function h_v is a "message function" that maps Alice's input set $\mathcal{X}$ to a vertex in V that v has an edge to. Similarly, for all $v \in V_B$, the function h_v maps Bob's input set $\mathcal{Y}$ to a vertex in V that v has an edge to. We often conflate the two and write $h_v : \mathcal{X} \times \mathcal{Y} \to V$ with the understanding that the second argument is redundant if $v \in V_A$ and the first argument is redundant if $v \in V_B$. Finally, for all $v \in V_O$, the value $o_v \in \mathcal{O}$ is the output value for the vertex v. We will assume that vertices in V_O do not have any out-edges, all other vertices have at least one outgoing edge (as otherwise the message functions will not be defined) and rt does not have any incoming edges. We define the size of Π to be $|V|$, the number of nodes in Π, and the depth of Π to be the length of the longest path in G starting from rt. We use $\|\Pi\|$ to denote the size of Π.

As mentioned above, edges in a **dag**-protocol go "down" from the root to the leaves, and are thus opposite to our convention for circuits. We are explicit about the direction in our exposition whenever there is room for ambiguity.

Execution of a dag-protocol

A **dag**-protocol Π as above is executed by start with inputs $x \in \mathcal{X}$, $y \in \mathcal{Y}$ and a vertex $v = \mathsf{rt}$, and continuing as follows: If $v \in V_A \cup V_B$ then the execution simply uses the message function and updates the current vertex to $h_v(x, y)$[9]. Otherwise, we have $v \in V_O$, and the execution terminates with the output o_v. As this output is determined by x and y, we will denote it using $\Pi(x, y)$. For a search problem $S \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{O}$, we say that Π solves S if for all $x \in \mathcal{X}$, $y \in \mathcal{Y}$, we have $(x, y, \Pi(x, y)) \in S$.

3.2 Rectangular Correctness & Equivalence of Circuits and dag-protocols

The models of Boolean circuits and **dag**-protocols described above have a deep connection. To understand this connection, we first define the notion of rectangular correctness for **dag**-protocols.

▶ **Definition 3** (Rectangles for **dag**-protocols). *Let Π be a* **dag***-protocol with input sets $\mathcal{X}$, $\mathcal{Y}$, and output set $\mathcal{O}$. We inductively define the (combinatorial) rectangle associated with each vertex. For the root rt, define the rectangle to $R_{\mathsf{rt}} = \mathcal{X} \times \mathcal{Y}$. For a node $v \neq \mathsf{rt}$, define the (combinatorial) rectangle $R_v \subseteq \mathcal{X} \times \mathcal{Y}$ to be the smallest rectangle containing $\bigcup_{u:(u,v)\in E}\{(x, y) \in R_u \mid h_u(x, y) = v\}$.*

For a combinatorial rectangle $R \subseteq \mathcal{X} \times \mathcal{Y}$, we will use R_A to denote the projection of the rectangle on the set $\mathcal{X}$ and R_B to denote the projection on the set $\mathcal{Y}$. As we defined R_v to be the smallest rectangle in Definition 3, we have the following observation.

▶ **Observation 4.** *Let Π be a* **dag***-protocol with input sets $\mathcal{X}$, $\mathcal{Y}$, and output set $\mathcal{O}$ and $\{R_v\}_{v\in V}$ be the associated rectangles. For all $v \neq \mathsf{rt}$ and all $x \in (R_v)_A$, there exists a $u \in V$ such that $(u, v) \in E$ and $y \in \mathcal{Y}$ such that $(x, y) \in R_u$ and $h_u(x, y) = v$. An analogous claim holds with the roles of x and y reversed.*

Additionally, observe that for all vertices v, the rectangle R_v contains all the pairs (x, y) such that the execution of Π goes to v on inputs x and y. Thus, the following notion of rectangular correctness is stronger than the "standard" notion of correctness.

▶ **Definition 5** (Rectangular Correctness). *Let Π be a* **dag***-protocol with inputs sets $\mathcal{X}$, $\mathcal{Y}$, and output set $\mathcal{O}$, and $\{R_v\}_{v\in V}$ be the associated rectangles. For a search problem $S \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{O}$, we say that Π solves S with rectangular correctness if for all $v \in V_O$ and all $(x, y) \in R_v$, we have $(x, y, o_v) \in S$.*

We are now ready to state the equivalence between circuits and **dag**-protocols formally. This theorem can also be found in [7]. We include a proof for completeness.

▶ **Theorem 6** ([25, 29]). *Let $f : \{0, 1\}^n \to \{0, 1\}$ be a Boolean function.*

1. *For any circuit C that computes f, there is a* **dag***-protocol Π with the same size and depth as C that solves KW_f with rectangular correctness.*
2. *For any* **dag***-protocol Π that solves KW_f with rectangular correctness, there is a circuit C with $\|C\| \leq \|\Pi\|$ that computes f.*

[9] Recall that $h_v(x, y)$ is independent of y if $v \in V_A$ and is independent of x if $v \in V_B$.

Proof. We only prove Item 1 as Item 2 is implied by Theorem 7 proved later. Fix a circuit C and define a **dag**-protocol Π for KW_f by setting the graph G to be the same graph as C except that the directions of all the edges are reversed. Let V_A be the set of all $\vee$ gates in C, V_B be the set of all the $\wedge$ gates in C and V_O be the set of all the inputs gates. Let rt be the output gate of C. For all $v \in V_O$, we define o_v to be the variable that (possibly after negation) is input at that gate. It remains to define the message transmission functions $\{h_v\}_{v \in V_A \cup V_B}$.

Fix $v \in V_A \cup V_B$ and recall that the input set $\mathcal{X} = f^{-1}(1)$ and $\mathcal{Y} = f^{-1}(0)$. For $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, we define $h_v(x, y)$ as follows: If $v \in V_A$, then $h_v(x, y)$ is just a function of x. If $f_{C,v}(x) = 1$, then, as v is a $\vee$ gate[10], by Equation (1), there exists a neighbor u (in-neighbor in C and out-neighbor in G) of v such that $f_{C,u}(x) = 1$, and we set $h_v(x, y)$ to be the lexicographically smallest such neighbor u. Otherwise, we set $h_v(x, y)$ to be the lexicographically smallest neighbor u of v. Observe that such a neighbor always exists as v is not an input gate. Similarly, if $v \in V_B$, then $h_v(x, y)$ is just a function of y. If $f_{C,v}(y) = 0$, then, as v is a $\wedge$ gate, by Equation (1), there exists a neighbor u (in-neighbor in C and out-neighbor in G) of v such that $f_{C,u}(y) = 0$, and we set $h_v(x, y)$ to be the lexicographically smallest such neighbor u. Otherwise, we set $h_v(x, y)$ to be the lexicographically smallest neighbor u of v. As before, such a neighbor always exists as v is not an input gate.

We clearly have $\|\Pi\| = \|C\|$ and we just have to show that Π solves KW_f with rectangular correctness. By Definition 5, we have to show that, if $\{R_v\}_{v \in V}$ are the rectangles associated with Π as in Definition 3, then, for all $v \in V_O$ and all $(x, y) \in R_v$, we have $(x, y, o_v) \in \mathsf{KW}_f$. In fact, we will show the stronger statement that for all $v \in V$ and all $(x, y) \in R_v$, we have:

$$f_{C,v}(x) = 1 \qquad \text{and} \qquad f_{C,v}(y) = 0.$$

This is indeed stronger, as if $v \in V_O$, then Equation (1) says that $f_{C,v}$ is just the (possibly negated) input variable o_v implying that $(x, y, o_v) \in \mathsf{KW}_f$. We show this by induction starting from the root rt downwards to the nodes reachable from it[11]. For the root rt, we use the fact that it corresponds to the output gate of C implying that $f_{C,\mathsf{rt}} = f$. The result follows as $\mathcal{X} = f^{-1}(1)$ and $\mathcal{Y} = f^{-1}(0)$.

We now show it for $v \neq \mathsf{rt}$ assuming it holds for all u that have edges to v in Π (equivalently, all u that v has an edge to in the circuit C). Let $(x, y) \in R_v$ be arbitrary. We only show that $f_{C,v}(x) = 1$ as the proof that $f_{C,v}(y) = 0$ is analogous. As $(x, y) \in R_v$, we have by Observation 4 that there exists u that has an edge to v in Π and a $y' \in \mathcal{Y}$ such that $(x, y') \in R_u$ and $h_u(x, y') = v$. Applying the induction hypothesis on u, it follows that $f_{C,u}(x) = 1$. Now, if $u \in V_\wedge$, then the fact that v has an edge to u in C together with Equation (1) implies that $f_{C,v}(x) = 1$, as desired. On the other hand, if $u \in V_\vee$, then, use the fact that $h_u(x, y') = v$ to get that $f_{C,v}(x) = 1$, finishing the proof. ◀

3.3 Error Models

Circuits

We consider short-circuit errors. Let C be a Boolean circuit as above. An error pattern for C is defined by a function $e : V \to V \cup \{*\}$ satisfying the property that $e(v)$ maps v to an in-neighbor of v, *i.e.* to a neighbor of v that is "below" it, or to $*$, for all $v \in V$. In particular,

[10] We abuse notation and use v to denote both the vertices in G and the corresponding vertices in C.
[11] Observe that if v is not reachable from rt, then Definition 3 says that $R_v = \emptyset$ and there is nothing to show.

if $v \notin V_\wedge \cup V_\vee$, we have $e(v) = *$. It will often be convenient to separate e into two functions and write $e = (e_\vee, e_\wedge)$, where $e_\vee$ is the function e restricted to the vertices in $V_\vee$ and $e_\wedge$ is the function e restricted to the vertices in $V_\wedge$. Let C be a circuit and e be an error pattern for C. Let $v \in C$ be a gate. We define how computation takes place in C in the presence of errors e by inductively defining the functions $f_{C,e,v}$ computed at each gate v. We have[12]:

$$f_{C,e,v} = \begin{cases} z, & \text{if } v \text{ is an input gate with label } z \\ \bigvee_{u:(u,v) \text{ is an edge in } C} f_{C,e,u}, & \text{if } v \in V_\vee \text{ and } e(v) = * \\ \bigwedge_{u:(u,v) \text{ is an edge in } C} f_{C,e,u}, & \text{if } v \in V_\wedge \text{ and } e(v) = * \\ f_{C,e,e(v)}, & \text{if } e(v) \neq * \end{cases}. \tag{2}$$

We omit v from the notation if v is the output gate of C. Let C be a circuit and $\mathcal{E}$ be a set of error patterns for C. For a function f, we say that C computes f despite $\mathcal{E}$, if for all $e \in \mathcal{E}$, we have $f_{C,e} = f$. We say that $\mathcal{E}$ is rectangular if viewing every $e \in \mathcal{E}$ as a pair $e = (e_\vee, e_\wedge)$ gives a combinatorial rectangle. When this happens, we use $\mathcal{E}_\vee$ to denote the projection of this rectangle on the first coordinate and $\mathcal{E}_\wedge$ to denote the projection on the second coordinate (thus, $\mathcal{E} = \mathcal{E}_\vee \times \mathcal{E}_\wedge$)

We now define the set of error patterns that we work with. Let $\Theta \geq 0$ be an integer parameter. For a circuit C, define the set $\mathcal{E}_\Theta(C)$ to be the set of all error patterns e such that for any path in C that starts at an input gate and ends the output gate, at most Θ gates $v \in V_\wedge$ on the path satisfy $e(v) \neq *$ and at most Θ gates $v \in V_\vee$ on the path satisfy $e(v) \neq *$. Observe that the set $\mathcal{E}_\Theta(C)$ is rectangular for all $\Theta \geq 0$.

dag-protocols

Let Π be a **dag**-protocol as above. An error pattern ξ for Π is defined by a function $\xi : V \to V \cup \{*\}$ satisfying the property that $\xi(v)$ maps v to an out-neighbor of v, *i.e.* to a neighbor of v that is "below" it[13], or to $*$. In particular, if $v \in V_O$, we have $\xi(v) = *$. It will often be convenient to separate ξ into two functions $\xi = (\xi_A, \xi_B)$, where ξ_A is the function ξ restricted to the vertices in V_A, and ξ_B is the function ξ restricted to the vertices in V_B. The error pattern ξ affects the execution of Π as follows: If the current vertex is v and $\xi(v) = *$, the execution proceeds as before. On the other hand, if $\xi(v) \neq *$ (observe that this can only happen if $v \in V_A \cup V_B$), the execution updates the current vertex to $\xi(v)$ instead of updating using the function h_v as before.

Let Π be a **dag**-protocol and Ξ be a set of error patterns for Π. Similarly to before, we say that Ξ is rectangular if viewing every $\xi \in \Xi$ as a pair $\xi = (\xi_A, \xi_B)$ gives a combinatorial rectangle. When this happens, we use Ξ_A to denote the projection of this rectangle on the first coordinate and Ξ_B to denote the projection on the second coordinate (thus, $\Xi = \Xi_A \times \Xi_B$). For a rectangular Ξ, define the **dag**-protocol Π_Ξ to be the same as Π except that the input sets are now $\mathcal{X} \times \Xi_A$ and $\mathcal{Y} \times \Xi_B$ and the message functions $h_{\Xi,v}$ are defined as (recall that $\xi = (\xi_A, \xi_B)$):

$$h_{\Xi,v}((x, \xi_A), (y, \xi_B)) = \begin{cases} h_v(x, y), & \text{if } \xi(v) = * \\ \xi(v), & \text{if } \xi(v) \neq * \end{cases}. \tag{3}$$

[12] Recall that the label of an input gate is either an input variable or the negation of an input variable.
[13] Recall that our conventions for the directions of edges in circuits is opposite that in **dag**-protocols.

For a search problem $S \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{O}$ and a rectangular Ξ, we say that Π solves S with rectangular correctness despite Ξ if Π_Ξ solves S_Ξ with rectangular correctness, where $S_\Xi \subseteq (\mathcal{X} \times \Xi_A) \times (\mathcal{Y} \times \Xi_B) \times \mathcal{O}$ is the search problem satisfying $((x, \xi_A), (y, \xi_B), o) \in S_\Xi \iff (x, y, o) \in S$ for all values of x, y, o, ξ_A, ξ_B.

We now define the set of error patterns that we work with. Let $\Theta \geq 0$ be an integer parameter. For a **dag**-protocol Π, define the set $\Xi_\Theta(\Pi)$ to be the set of all error patterns ξ such that for any path in G that starts at rt and ends at a node in V_O, at most Θ nodes $v \in V_A$ on the path satisfy $\xi(v) \neq *$ and at most Θ nodes $v \in V_B$ on the path satisfy $\xi(v) \neq *$. Observe that the set $\Xi_\Theta(\Pi)$ is rectangular for all $\Theta \geq 0$. For convenience, we will often abbreviate $\Pi_{\Xi_\Theta(\Pi)}$ to Π_Θ.

3.4 Connecting the Error Models

We now show that error resilient **dag**-protocols are stronger than error-resilient circuits. Observe that the theorem below implies Item 2 of Theorem 6 as can be seen by setting $\Theta = 0$. Additionally, note that the theorem below differs from the analogous theorem in [7] in that the number of allowed errors Θ is the same for every path, while in [7], it was a constant fraction of the length of the path. Having it as a constant fraction creates some minor complications while trimming the "empty edges" (in the first paragraph of the proof), that were overlooked in [7]. These complications do not affect the rest of the proof of [7], as [7] actually works even when the allowed number of errors is the same for every path (and equals a constant fraction of the length of the *longest* path). Nonetheless, we flesh out the details of trimming for our model formally in the full version. We also mention that having the same number of errors on all the paths, irrespective of their length, is a stronger error model than having the number of errors on each path be proportional to its length.

▶ **Theorem 7.** *Let $\Theta \geq 0$ be given and $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. For any* **dag***-protocol Π that solves KW_f with rectangular correctness despite $\Xi_\Theta(\Pi)$, there is a circuit C with size and depth at most that of Π that computes f despite $\mathcal{E}_\Theta(C)$.*

Proof. Fix a protocol Π that solves KW_f with rectangular correctness despite $\Xi_\Theta(\Pi)$. We can assume without loss of generality that there are no empty edges in Π, as defined in the full version. Equivalently, if $\{R_{\Theta,v}\}_{v \in V}$ are the rectangles associated with Π_Θ according to Definition 3 and $\{h_{\Theta,v}\}_{v \in V_A \cup V_B}$ are the message functions, for all edges $e = (u, v) \in E$ we have:

$$R_{\Theta,u} = \emptyset \quad \text{or} \quad \exists ((x, \xi_A), (y, \xi_B)) \in R_{\Theta,u} : \qquad h_{\Theta,u}((x, \xi_A), (y, \xi_B)) = v. \tag{4}$$

As Π that solves KW_f with rectangular correctness despite $\Xi_\Theta(\Pi)$, we have that Π_Θ solves $\mathsf{KW}_{f,\Xi_\Theta(\Pi)}$ with rectangular correctness. From Definition 5, we get that for all $v \in V_O$ and all $((x, \xi_A), (y, \xi_B)) \in R_{\Theta,v}$, we have $((x, \xi_A), (y, \xi_B), o_v) \in \mathsf{KW}_{f,\Xi_\Theta(\Pi)}$. Simplifying using the definition of $\mathsf{KW}_{f,\Xi_\Theta(\Pi)}$, we get:

$$\forall v \in V_O, ((x, \xi_A), (y, \xi_B)) \in R_{\Theta,v} : \quad x_{o_v} \neq y_{o_v}. \tag{5}$$

We now define the circuit C. The gates in the circuit C are exactly the nodes in V and we abuse notation slightly by using u, v *etc.* to refer to both. For any edge $(u, v) \in E$, we add the flipped edge (v, u) to the circuit C. We define all gates in V_A to be $\vee$ gates and all gates in V_B to be $\wedge$ gates while rt is chosen as the output gate. The remaining gates are in V_O are input gates whose labels are defined next. As $R_{\Theta,v}$ in Equation (5) is a combinatorial rectangle, we get that there exists $b \in \{0,1\}$ such that

$$\forall ((x, \xi_A), (y, \xi_B)) \in R_{\Theta,v} : \quad x_{o_v} = b \quad \text{and} \quad y_{o_v} = \overline{b}.$$

If $b = 1$, we label the input gate v by the variable o_v unnegated, and we label it by the negated variable $\overline{o_v}$ otherwise. This finishes the description of C. As the claim about the size $\|C\|$ is straightforward, we focus on showing that C computes f despite $\mathcal{E}_\Theta(C)$. As both C and Π have the same graph upto the directions of the edges, observe that any error pattern for C can also be seen as a pattern for Π and that $\mathcal{E}_\Theta(C) = \Xi_\Theta(\Pi)$. Because of this, we interpret any error pattern $\xi = (\xi_A, \xi_B)$ as an error pattern $e = (e_\vee, e_\wedge)$ for C and we ξ and subscripts A and B to refer to both. We also define the error patterns $*_A$ and $*_B$ to be those that map all gates in V_A (equivalently, $V_\vee$) and V_B (equivalently, $V_\wedge$) respectively to $*$. To show that C computes f despite $\mathcal{E}_\Theta(C)$, we show inductively that for all $u \in V$, we have:

$$\forall((x,\xi_A),(y,\xi_B)) \in R_{\Theta,u}: \quad f_{C,(\xi_A,*_B),u}(x) = 1 \quad \text{and} \quad f_{C,(*_A,\xi_B),u}(y) = 0. \tag{6}$$

This suffices as plugging $v = \mathsf{rt}$ and using the definition of $R_{\Theta,\mathsf{rt}} = \left(f^{-1}(1) \times (\Xi_\Theta(\Pi))_A\right) \times \left(f^{-1}(0) \times (\Xi_\Theta(\Pi))_B\right)$ from Definition 3, we get that:

$$\forall((x,\xi_A),(y,\xi_B)) \in R_{\Theta,\mathsf{rt}}: \quad f_{C,(\xi_A,*_B)}(x) = f(x) \quad \text{and} \quad f_{C,(*_A,\xi_B)}(y) = f(y).$$

As short-circuiting a gate in $V_\wedge$ can never change the output from 1 to 0 and short-circuiting a gate in $V_\vee$ can never change the output from 0 to 1, we get that:

$$\forall((x,\xi_A),(y,\xi_B)) \in R_{\Theta,\mathsf{rt}}: \quad f_{C,\xi}(x) = f(x) \quad \text{and} \quad f_{C,\xi}(y) = f(y).$$

Thus, we get that $f_{C,\xi} = f$ for all $\xi \in \Xi_\Theta(\Pi)$. As $\mathcal{E}_\Theta(C) = \Xi_\Theta(\Pi)$, we get that C computes f despite $\mathcal{E}_\Theta(C)$, as desired. Having shown that Equation (6) is sufficient, we now prove that Equation (6) holds by induction. For the base case, when $u \in V_O$, this follows by our labels and Equations (2) and (5). We show the results for $u \notin V_O$ assuming it holds for all v that have edges to u in C. If $R_{\Theta,u} = \emptyset$, there is nothing to show, so we assume otherwise. As the proof in the other case is similar, assume without loss of generality that $u \in V_A$. From Equation (4), we get that for all v that have edges to u in C, we have that there exists $((x,\xi_A),(y,\xi_B)) \in R_{\Theta,u}$ such that $h_{\Theta,u}((x,\xi_A),(y,\xi_B)) = v$. As $u \in V_A$, the message function is determined by the first argument and we get from Definition 3 that:

$$(R_{\Theta,u})_A \subseteq \bigcup_{v:(v,u) \text{ is an edge in } C} (R_{\Theta,v})_A \quad \text{and} \quad (R_{\Theta,u})_B \subseteq \bigcap_{v:(v,u) \text{ is an edge in } C} (R_{\Theta,v})_B. \tag{7}$$

We now show Equation (6). Fix $((x,\xi_A),(y,\xi_B)) \in R_{\Theta,u}$.

- **Showing that $f_{C,(\xi_A,*_B),u}(x) = 1$ when $\xi(u) \neq *$:** By Equation (3), we get that $h_{\Theta,u}((x,\xi_A),(y,\xi_B)) = \xi(u)$ implying from Definition 3 that $((x,\xi_A),(y,\xi_B)) \in R_{\Theta,\xi(u)}$. By the induction hypothesis on $\xi(u)$, we get that $f_{C,(\xi_A,*_B),\xi(u)}(x) = 1$. From Equation (2), we get:

$$f_{C,(\xi_A,*_B),u}(x) = f_{C,(\xi_A,*_B),\xi(u)}(x) = 1.$$

- **Showing that $f_{C,(\xi_A,*_B),u}(x) = 1$ when $\xi(u) = *$:** We have from $((x,\xi_A),(y,\xi_B)) \in R_{\Theta,u}$ that $(x,\xi_A) \in (R_{\Theta,u})_A$. From Equation (7), there exists an in-neighbor v' of u in C and $(y_{v'},\xi_{B,v'})$ such that $((x,\xi_A),(y_{v'},\xi_{B,v'})) \in R_{\Theta,v'}$. By the induction hypothesis on v', we have $f_{C,(\xi_A,*_B),v'}(x) = 1$. From Equation (2), we get:

$$f_{C,(\xi_A,*_B),u}(x) = \bigvee_{v:(v,u) \text{ is an edge in } C} f_{C,(\xi_A,*_B),v}(x) = 1.$$

- **Showing that $f_{C,(*_A,\xi_B),u}(y) = 0$:** We have from $((x,\xi_A),(y,\xi_B)) \in R_{\Theta,u}$ that $(y,\xi_B) \in (R_{\Theta,u})_B$. It follows from Equation (7) that for all in-neighbors v of u in C, we have that there exists $(x_v,\xi_{A,v})$ such that $((x_v,\xi_{A,v}),(y,\xi_B)) \in R_{\Theta,v}$. From the induction hypothesis on v, we get that $f_{C,(*_A,\xi_B),v}(y) = 0$ for all such v. From Equation (2), we get:

$$f_{C,(*_A,\xi_B),u}(y) = \bigvee_{v:(v,u) \text{ is an edge in } C} f_{C,(*_A,\xi_B),v}(y) = 0. \qquad \blacktriangleleft$$

3.5 Our Results

We are now ready to state our main result formally. We show that:

▶ **Theorem 8** (Formal version of Theorem 1). *Let $c > 1$, $n > 0$ be integers and $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. Let C be a Boolean circuit of size $\|C\| = s$ and depth*[14] *$d < (\log n)^{c-1}$ that computes f. There exists $\Theta \leq s^{\mathcal{O}(c \log c)}$ and a Boolean circuit C' with size at most $s^{\mathcal{O}(c \log c)}$ and depth $\mathcal{O}(c) \cdot \Theta$ that computes f despite $\mathcal{E}_\Theta(C')$.*

To prove Theorem 8, we need the following result about **dag**-protocols, which forms the technical core of Theorem 8.

▶ **Theorem 9** (Formal version of Theorem 2). *Let S be a search problem and Π be a* **dag***-protocol of size s and depth d that solves S with rectangular correctness. Let $c > 1$ be an integer satisfying*[15] *$d < (\log s)^{c-1}$. There exists $\Theta \leq s^{\mathcal{O}(c \log c)}$ (defined in the full version) and a* **dag***-protocol Π' (as defined in the full version) with size $s' = s^{\mathcal{O}(c \log c)}$ and depth $\mathcal{O}(c) \cdot \Theta$ that solves S with rectangular correctness despite $\Xi_\Theta(\Pi')$.*

We prove Theorem 9 in the full version, but use it here to prove Theorem 8.

Proof of Theorem 8 assuming Theorem 9. Fix a circuit C as in the theorem statement. Applying Theorem 6, we get that there exists a **dag**-protocol Π with size s and depth d that solves KW_f with rectangular correctness. Applying Theorem 9, we get that there exists $\Theta \leq s^{\mathcal{O}(c \log c)}$ and a **dag**-protocol Π' with size $s' = s^{\mathcal{O}(c \log c)}$ and depth $\mathcal{O}(c) \cdot \Theta$ that solves KW_f with rectangular correctness despite $\Xi_\Theta(\Pi')$. Now, applying Theorem 7 on Π', we get that there exists a circuit C' with size at most $s^{\mathcal{O}(c \log c)}$ and depth at most $\mathcal{O}(c) \cdot \Theta$ that computes f despite $\mathcal{E}_\Theta(C')$, as desired. ◀

References

1. Mirza Ahad Baig, Suvradip Chakraborty, Stefan Dziembowski, Małgorzata Gałązka, Tomasz Lizurej, and Krzysztof Pietrzak. Efficiently testable circuits. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, 2023. doi:h10.4230/LIPIcs.ITCS.2023.10.
2. Mirza Ahad Baig, Suvradip Chakraborty, Stefan Dziembowski, Małgorzata Gałązka, Tomasz Lizurej, and Krzysztof Pietrzak. Efficiently testable circuits without conductivity. In *Theory of Cryptography Conference*, 2023. doi:10.1007/978-3-031-48621-0_5.
3. Maria Luisa Bonet and Samuel R. Buss. Size-depth tradeoffs for boolean fomulae. *Information Processing Letters*, 49(3):151–155, 1994. doi:10.1016/0020-0190(94)90093-0.
4. Mark Braverman, Klim Efremenko, Ran Gelles, and Michael A. Yitayew. Optimal short-circuit resilient formulas. In *Computational Complexity Conference (CCC)*, volume 137, pages 10:1–10:22, 2019. doi:10.4230/LIPICS.CCC.2019.10.

[14] The "−1" in this expression is for technical convenience.
[15] The "−1" in this expression is for technical convenience.

5 T.-H. Hubert Chan, Zhibin Liang, Antigoni Polychroniadou, and Elaine Shi. Small memory robust simulation of client-server interactive protocols over oblivious noisy channels. In *Symposium on Discrete Algorithms (SODA)*, pages 2349–2365, 2020. doi:10.1137/1.9781611975994.144.

6 Roland L'vovich Dobrushin and SI Ortyukov. Upper bound on the redundancy of self-correcting arrangements of unreliable functional elements. *Problemy Peredachi Informatsii*, 13(3):56–76, 1977.

7 Klim Efremenko, Bernhard Haeupler, Yael Tauman Kalai, Pritish Kamath, Gillat Kol, Nicolas Resch, and Raghuvansh R. Saxena. Circuits resilient to short-circuit errors. In *54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, 2022.

8 William S. Evans and Nicholas Pippenger. On the maximum tolerable noise for reliable computation by formulas. *IEEE Transactions on Information Theory*, 44(3):1299–1305, 1998. doi:10.1109/18.669417.

9 William S. Evans and Leonard J. Schulman. Signal propagation and noisy circuits. *IEEE Transactions on Information Theory*, 45(7):2367–2373, 1999. doi:10.1109/18.796377.

10 William S. Evans and Leonard J. Schulman. On the maximum tolerable noise of k-input gates for reliable computation by formulas. *IEEE Transactions on Information Theory*, 49:3094–3098, 2003. doi:10.1109/TIT.2003.818405.

11 Tomás Feder. Reliable computation by networks in the presence of noise. *IEEE Transactions on Information Theory*, 35(3):569–571, 1989. doi:10.1109/18.30978.

12 Péter Gács and Anna Gál. Lower bounds for the complexity of reliable boolean circuits with noisy gates. *IEEE Transactions on Information Theory*, 40(2):579–583, 1994. doi:10.1109/18.312190.

13 Anna Gál. Lower bounds for the complexity of reliable boolean circuits with noisy gates. In *Foundations of Computer Science (FOCS)*, pages 594–601, 1991. doi:10.1109/SFCS.1991.185424.

14 Anna Gál and Mario Szegedy. Fault tolerant circuits and probabilistically checkable proofs. In *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*, pages 65–73, 1995. doi:10.1109/SCT.1995.514728.

15 Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. In *Symposium on Theory of Computing (STOC)*, pages 902–911, 2018. doi:10.1145/3188745.3188838.

16 Ran Gelles. Coding for interactive communication: A survey. *Foundations and Trends in Theoretical Computer Science*, 13(1–2):1–157, 2017. doi:10.1561/0400000079.

17 Bruce E. Hajek and Timothy Weller. On the maximum tolerable noise for reliable computation by formulas. *IEEE Transactions on Information Theory*, 37(2):388–391, 1991. doi:10.1109/18.75261.

18 Yael Tauman Kalai, Allison B. Lewko, and Anup Rao. Formulas resilient to short-circuit errors. In *Foundations of Computer Science (FOCS)*, pages 490–499, 2012. doi:10.1109/FOCS.2012.69.

19 Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In *Symposium on Theory of Computing (STOC)*, pages 539–550, 1988. doi:10.1145/62212.62265.

20 Daniel J. Kleitman, Frank Thomson Leighton, and Yuan Ma. On the design of reliable boolean circuits that contain partially unreliable gates. *Journal of Computer and System Sciences*, 55(3):385–401, 1997. doi:10.1006/JCSS.1997.1531.

21 Jan Krajícek. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *Journal of Symbolic Logic*, 62(2):457–486, 1997. doi:10.2307/2275541.

22 Nicholas Pippenger. On networks of noisy gates. In *Foundations of Computer Science (FOCS)*, pages 30–38, 1985. doi:10.1109/SFCS.1985.41.

23 Nicholas Pippenger. Reliable computation by formulas in the presence of noise. *IEEE Transactions on Information Theory*, 34(2):194–197, 1988. doi:10.1109/18.2628.

24 Pavel Pudlák. On extracting computations from propositional proofs (a survey). In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010)*, volume 8, pages 30–41, 2010. doi:10.4230/LIPICS.FSTTCS.2010.30.

25 Alexander Razborov. Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izvestiya of the RAN*, pages 201–224, 1995.

26 Leonard J Schulman. Communication on noisy channels: A coding theorem for computation. In *Foundations of Computer Science (FOCS)*, pages 724–733. IEEE, 1992. doi:10.1109/SFCS.1992.267778.

27 Leonard J. Schulman. Deterministic coding for interactive communication. In *Symposium on Theory of Computing (STOC)*, pages 747–756, 1993. doi:10.1145/167088.167279.

28 Leonard J Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996. doi:10.1109/18.556671.

29 Dmitry Sokolov. Dag-like communication and its applications. In *Proceedings of the 12th Computer Science Symposium in Russia (CSR)*, pages 294–307. Springer, 2017. doi:10.1007/978-3-319-58747-9_26.

30 John von Neumann. Probabilistic logics and the synthesis of reliable organisms from unreliable components. *Automata Studies*, pages 43–98, 1956.

Simple Norm Bounds for Polynomial Random Matrices via Decoupling

Madhur Tulsiani
Toyota Technological Institute at Chicago, IL, USA

June Wu
University of Chicago, IL, USA

Abstract

We present a new method for obtaining norm bounds for random matrices, where each entry is a low-degree polynomial in an underlying set of independent real-valued random variables. Such matrices arise in a variety of settings in the analysis of spectral and optimization algorithms, which require understanding the spectrum of a random matrix depending on data obtained as independent samples.

Using ideas of decoupling and linearization from analysis, we show a simple way of expressing norm bounds for such matrices, in terms of matrices of lower-degree polynomials corresponding to derivatives. Iterating this method gives a simple bound with an elementary proof, which can recover many bounds previously required more involved techniques.

2012 ACM Subject Classification Mathematics of computing → Probability and statistics; Theory of computation → Theory and algorithms for application domains

Keywords and phrases Matrix Concentration, Decoupling, Graph Matrices

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.91

Funding *Madhur Tulsiani*: NSF grants CCF-1816372 and CCF-2326685.

1 Introduction

For fixed matrices $\mathbf{C_1}, \ldots, \mathbf{C_n}$ and independent scalar random variables $x_1, \ldots, x_n$, consider the problem of analyzing the random matrix

$$\mathbf{M} \;=\; \mathbf{C_1} \cdot x_1 + \cdots + \mathbf{C_n} \cdot x_n \,.$$

Note that the entries of the random matrix $\mathbf{M}$ are not necessarily independent, but are (possibly correlated) linear functions of the independent random variables $x_1, \ldots, x_n$. Such matrices, which arise in a variety of applications in algorithms, statistics, and numerical linear algebra, can often be shown as being concentrated around the mean, using a rich selection of matrix deviation inequalities [34]. Moreover, when the random variables $x_1, \ldots, x_n$ are Gaussian, recent breakthrough results have obtained even sharper concentration guarantees depending on the structure of the matrices $\mathbf{C_1}, \ldots, \mathbf{C_n}$ [5, 9] already leading to applications in discrepancy theory [6] and quantum information theory [20].

In contrast to the above random matrices which are linear functions of independent random variables, several recent applications in spectral algorithms and lower bounds for statistical problems require understanding the (expected) norms of random matrices of the form

$$\mathbf{F} \;=\; \sum_{\substack{S \subseteq [n] \\ |S| \leq d}} \mathbf{C}_S \cdot \prod_{i \in S} x_i \,.$$

The above random matrix, which denote as a matrix-valued function $\mathbf{F}(\mathbf{x})$, is a (multilinear, in the example above) *low-degree polynomial* in the vector of random variables $\mathbf{x} = (x_1, \ldots, x_n)$.


© Madhur Tulsiani and June Wu;
licensed under Creative Commons License CC-BY 4.0



Such matrix-valued polynomial functions arise naturally in a variety of applications, including algorithms for tensor decomposition and completion [13, 12, 18, 10], orbit recovery [24], power-sum decompositions and learning of Gaussian mixtures [4], and lower bounds for Sum-of-Squares hierarchies [2, 7, 16, 28]. In general, such matrices are often an important technical component in the analysis of spectral algorithms, which require understanding the spectrum of some random matrix depending on data obtained as independent samples. Given the expressive power of polynomials, one can often write the matrix entries as low-degree polynomials in the data.

In several applications above, one is interested in obtaining bounds on the spectral norm of the deviation matrix $\mathbf{F}(\mathbf{x}) - \mathbb{E}\,\mathbf{F}(\mathbf{x})$, which hold with high probability over the choice of $\mathbf{x}$. Although matrix-deviation inequalities for nonlinear random matrices have been a subject of active research in recent work [26, 22, 3, 14, 15], the analyses in the applications above have often needed to rely on estimating matrix norms via direct computations of trace moments. While these do yield sharp bounds required for applications, they require somewhat intricate computations and ingenious combinatorial arguments tailored to the applications at hand.

Analyzing concentration via moment estimates. We consider concentration bounds on a *scalar* polynomial $f(\mathbf{x})$ with mean zero. Using Markov's inequality, this can be reduced to computing moment estimates, since

$$\mathbb{P}\left[|f(\mathbf{x})| \geq \lambda\right] \;=\; \mathbb{P}\left[\left(f(\mathbf{x})\right)^{2t} \geq \lambda^{2t}\right] \;\leq\; \lambda^{-2t} \cdot \mathbb{E}\left[\left(f(\mathbf{x})\right)^{2t}\right]$$

Note that while in some cases $\mathbb{E}[(f(\mathbf{x}))^{2t}]$ can be computed by direct expansion, it often involves an intricate analysis of the structure of terms with degrees growing with t, and therefore indirect methods may be more convenient. Methods for obtaining concentration of (scalar) polynomial functions have been the subject of a large body of work, including results by Kim and Vu [17], Latała [21], Schudy and Sviridenko [31, 32], Adamczak and Wolff [1], and Bobkov, Götze and Sambale [8]. A useful comparison to direct computation, is the method of Adamczak and Wolff, which is applicable to functions $f(\mathbf{x})$ of a broad range of random vectors $\mathbf{x}$ (with not necessarily independent coordinates) from a distribution obeying the Poincaré inequality. They obtain such moment bounds by recursive applications of the Poincaré inequality, reducing the computation of the moments of f to that of its derivatives. For a degree-d polynomial function f, one can consider its coefficients as forming a (symmetric) order-d tensor, and their results obtain bounds on the moments of f in terms of the norms of various lower-order tensors "flattening" of this coefficient tensor. Thus, the problem of estimating moments of the random variable $f(\mathbf{x})$, can be reduced in a black-box way, to understanding the norms of a small number (depending on d) of *deterministic* tensors. Any dependence on the problem structure can then be limited to understanding these norms, where often crude estimates can suffice.

The matrix analog of the Markov argument involves the Schatten-$2t$ norm $\|.\|_{2t}$, which is defined for a matrix $\mathbf{M}$ with non-zero singular values $\sigma_1, \ldots, \sigma_r$ as $\|\mathbf{M}\|_{2t}^{2t} := \sum_{j\in[r]} \sigma_j^{2t}$. For a function $\mathbf{F}$ with $\mathbb{E}[\mathbf{F}(Z)] = \mathbf{0}$, we have the following bound using Schatten norms.

$$\mathbb{P}\left[\sigma_1(\mathbf{F}) \geq \lambda\right] \;\leq\; \lambda^{-2t} \cdot \mathbb{E}\,\|\mathbf{F}\|_{2t}^{2t} \;=\; \lambda^{-2t} \cdot \mathbb{E}\,\mathrm{tr}\left[\left(\mathbf{F}(\mathbf{x})\mathbf{F}(\mathbf{x})^\intercal\right)^t\right]$$

Several applications requiring norm bounds for matrix-valued polynomial functions, start with the above inequality, and often rely on direct expansion of the trace for the matrix power $(\mathbf{F}(\mathbf{x})\mathbf{F}(\mathbf{x})^\intercal)^t$. Recall that when $\mathbf{A}$ is the adjacency matrix of a graph, $\mathrm{tr}(\mathbf{A}^{2t})$ can be interpreted as the number of cycles of length $2t$. Similarly, when working with matrices $\mathbf{F}(\mathbf{x})$

where each entry can be interpreted as arising from a combinatorial pattern, $(\mathbf{F}(\mathbf{x})\mathbf{F}(\mathbf{x})^\intercal)^t$ can be viewed as the number of copies of such patterns "chained" together in a certain way. Estimating the trace then amounts to estimating the (expected) number of such chainings [2].

Obtaining bounds on the $\mathbb{E}\,\|\mathbf{F}\|_{2t}^{2t}$ via this method requires first using problem structure to decompose $\mathbf{F}$ into matrices with such combinatorial patterns, and then using (ingenious) combinatorial arguments to understand the expected number of occurrences of such chains of patterns. The focus of this work is understanding alternative general methods for estimating such norm bounds, when the underlying problem may not necessarily have a combinatorial flavor, or when such decompositions may be hard to analyze.

Decoupling. Decoupling inequalities were developed in the study of U-statistics [27], multiple stochastic integration [23], and polynomial chaos [19], and have found important applications in applied mathematics, theoretical computer science, applied probability theory and statistics. Some examples include the study of compressed sensing [30], query complexity [25], the proof of Hanson-Wright inequality [35], learning mixture of Gaussians [11] and so on. Moreover, the inequalities are applicable in both scalar and matrix settings, and even more broadly in Banach spaces.

For a degree-d homogeneous *multilinear* polynomial $f(\mathbf{x})$ in n independent random variables, standard decoupling inequalities (see Section 3) can be used to obtain

$$\mathop{\mathbb{E}}_{\mathbf{x}}\left[(f(\mathbf{x}))^{2t}\right] \;\leq\; C_d^{2t}\cdot \mathop{\mathbb{E}}_{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(d)}}\left[\left(f(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(d)})\right)^{2t}\right],$$

where $f(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(d)})$ denotes the polynomial in $d\cdot n$ random variables obtained by replacing the d variables in each (ordered) monomial by the corresponding d coordinates coming from the d independent random vectors $\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(d)}$, and C_d is a constant depending on the degree d. One can now fix the vectors $\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(d-1)}$, and consider the expectation

$$\mathop{\mathbb{E}}_{\mathbf{x}^{(d)}}\left[\left(f(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(d)})\right)^{2t}\right] \;=\; \mathop{\mathbb{E}}_{\mathbf{x}^{(d)}}\left[\left(\sum_{i\in[n]} f_i(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(d-1)})\cdot x_i^{(d)}\right)^{2t}\right],$$

where we write $f(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(d)})$ as a linear form in $\mathbf{x}^{(d)}$ with coefficients $f_i(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(d-1)})$. The moments of such a linear form can be understood (for example) using the Khintchine inequality, which bounds it in terms of a polynomial depending only on $\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(d-1)}$.

Similarly, in the matrix case, we use decoupling inequalities to reduce the problem of estimating moments for degree-d homogeneous (and multilinear) matrix-valued polynomials, to that of estimating moments for *linear* matrix-valued functions. At this point, one can apply standard and well-known linear matrix concentration inequalities (we use the matrix Rosenthal inequality) which yield a bound in terms of a degree-$(d-1)$ matrix-valued polynomial function of the vectors $\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(d-1)}$. Iterating this process gives a simple method for obtaining norm bounds, applicable for *multilinear* polynomial functions on n independent random variables (with any reasonable distribution).

While this method does not immediately apply for non-multilinear polynomials, it suffices for many of the applications mentioned above. Moreover, for arbitrary polynomials in Gaussian random variables, we can also obtain a simple recursion as an immediate consequence of the more recent Poincaré inequalities of Huang and Tropp [15]. Together, these suffice for most applications of interest.

Methods and results: a technical overview

Let $x_1, \ldots, x_n$ be i.i.d real-valued random variables with $\mathbb{E}[x_i] = 0, \mathbb{E}[x_i^2] = 1$ and $|x_i| \leq L$ for each $i \in [n]$. Let $\mathcal{T}_n^d \subseteq [n]^d$ denote the set of ordered d-tuples $\mathbf{i} = (i_1, \ldots, i_d)$ with $i_1, \ldots, i_d$ all distinct. For a fixed sequence of (deterministic) matrices $\{\mathbf{A_i}\}_{\mathbf{i} \in \mathcal{T}_n^d} \subseteq \mathbb{R}^{d_1 \times d_2}$, we consider a matrix-valued multilinear polynomial function defined as

$$\mathbf{F}(\mathbf{x}) \;=\; \sum_{\mathbf{i} \in \mathcal{T}_n^d} \mathbf{A_i} \cdot \prod_{j \in [d]} x_{i_j} \,.$$

We write the polynomial in terms of ordered tuples for decoupling, but since the variables $x_1, \ldots, x_n$ commute, we can assume that for any permutation $\sigma : [d] \to [d]$ permuting the d coordinates of each tuple, $\mathbf{A_i} = \mathbf{A}_{\sigma(\mathbf{i})}$ (this is referred to as *permutation symmetry* of $\mathbf{F}$). We can use the decoupling inequalities from Section 3 to show that

$$\|\mathbf{F}(\mathbf{x})\|_{4t} \;\leq\; C_d \cdot \left\|\mathbf{F}(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(d)})\right\|_{4t} \;=\; C_d \cdot \left\|\sum_{\mathbf{i} \in \mathcal{T}_n^d} \mathbf{A_i} \cdot \prod_{j \in [d]} x_{i_j}^{(j)}\right\|_{4t} ,$$

where $\|\cdot\|_{4t}$ denotes the (expected) Schatten norm $\left(\mathbb{E}[\mathrm{tr}(\mathbf{F}(\mathbf{x})^\mathsf{T}\mathbf{F}(\mathbf{x}))^{2t}]\right)^{1/4t}$.

We remark that while the constant C_d is of the form d^d in general, this can be improved when the underlying set of indices $[n]$ has more structure. For example, when $[n]$ corresponds to the set of *pairs* in an base set $[r]$ and each of the monomials involves at most k elements of $[r]$ (has *index degree* k), it is possible to improve the constant to k^k. This is essentially the "vertex partitioning" argument used in works on graph matrices, and is discussed in Section 3 in the language of decoupling.

As mentioned earlier, we can now "linearize" the function $\mathbf{F}(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(d)})$ by fixing $\mathbf{x}^{(d)}, \ldots, \mathbf{x}^{(d-1)}$ and treating it only as a function of $\mathbf{x}^{(d)}$ i.e., we consider

$$\begin{aligned}
\mathbb{E}\left\|\sum_{\mathbf{i} \in \mathcal{T}_n^d} \mathbf{A_i} \cdot \prod_{j \in [d]} x_{i_j}^{(j)}\right\|_{4t}^{4t} &= \mathop{\mathbb{E}}_{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(d-1)}} \mathop{\mathbb{E}}_{\mathbf{x}^{(d)}} \left\|\sum_{k \in [n]} \left(\sum_{\substack{\mathbf{i} \in \mathcal{T}_n^d \\ i_d = k}} \mathbf{A_i} \cdot \prod_{j \in [d-1]} x_{i_j}^{(j)}\right) \cdot x_k^{(d)}\right\|_{4t}^{4t} \\
&= \mathop{\mathbb{E}}_{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(d-1)}} \mathop{\mathbb{E}}_{\mathbf{x}^{(d)}} \left\|\sum_{k \in [n]} \left(\partial_{x_k^{(d)}} \mathbf{F}\right) \cdot x_k^{(d)}\right\|_{4t}^{4t} .
\end{aligned}$$

To bound the inner expectation, we can now use the matrix Rosenthal inequality (Lemma 8) which says that for a collection of centered, independent random matrices $\{\mathbf{Y}_k\}$ with finite moments, we have

$$\mathbb{E}\left\|\sum_k \mathbf{Y}_k\right\|_{4t}^{4t} \leq (16t)^{3t} \cdot \left\{\left\|\left(\sum_k \mathbb{E}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\right)^{1/2}\right\|_{4t}^{4t} + \left\|\left(\sum_k \mathbb{E}\mathbf{Y}_k^\mathsf{T}\mathbf{Y}_k\right)^{1/2}\right\|_{4t}^{4t}\right\} + (8t)^{4t} \cdot \left(\sum_k \mathbb{E}\left\|\mathbf{Y}_k\right\|_{4t}^{4t}\right).$$

Taking $\mathbf{Y}_k = \partial_{x_k^{(d)}} \mathbf{F} \cdot x_k^{(d)}$ we can now compute the expectations (over $\mathbf{x}^{(d)}$) in the RHS as

$$\sum_k \mathbb{E}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T} = \left[\partial_{x_1^{(d)}}\mathbf{F} \quad \dots \quad \partial_{x_n^{(d)}}\mathbf{F}\right]\left[\partial_{x_1^{(d)}}\mathbf{F} \quad \dots \quad \partial_{x_n^{(d)}}\mathbf{F}\right]^\mathsf{T}$$

$$\sum_k \mathbb{E}\mathbf{Y}_k^\mathsf{T}\mathbf{Y}_k = \left[\partial_{x_1^{(d)}}\mathbf{F}^\mathsf{T} \quad \dots \quad \partial_{x_n^{(d)}}\mathbf{F}^\mathsf{T}\right]\left[\partial_{x_1^{(d)}}\mathbf{F}^\mathsf{T} \quad \dots \quad \partial_{x_n^{(d)}}\mathbf{F}^\mathsf{T}\right]^\mathsf{T}$$

$$\sum_k \mathbb{E}\left\|\mathbf{Y}_k\right\|_{4t}^{4t} = \mathbb{E}\left\|\begin{bmatrix} x_1^{(d)}\cdot\partial_{x_1^{(d)}}\mathbf{F} & & \\ & \ddots & \\ & & x_n^{(d)}\cdot\partial_{x_n^{(d)}}\mathbf{F}\end{bmatrix}\right\|_{4t}^{4t} \le L^{4t}\cdot\left\|\begin{bmatrix}\partial_{x_1^{(d)}}\mathbf{F} & & \\ & \ddots & \\ & & \partial_{x_n^{(d)}}\mathbf{F}\end{bmatrix}\right\|_{4t}^{4t}$$

Iterating the above argument, we can prove a bound in terms of a collection of partial derivative matrices $\mathbf{F}_{a,b,c}$ (see Section 2 for details).

Note that in the above definition, we do not specify the order of differentiation, which can be ignored due to the permutation symmetry of $\mathbf{F}$ and the fact that changing the order does not change the Schatten norms (see Section 2 and Section 4 for details). This argument yields the following bound for all homogeneous multilinear polynomial functions.

▶ **Theorem 1** (Restatement of Theorem 10). *Let $\mathbf{x} = \{x_i\}_{i=1}^n$ be a sequence of i.i.d random variables with $\mathbb{E}x_i = 0$, $\mathbb{E}x_i^2 = 1$ and $|x_i| \le L$ for all $1 \le i \le n$. Let $\{\mathbf{A_i}\}_{\mathbf{i}\in\mathcal{T}_n^d}$ be a multi-indexed sequence of deterministic matrices of the same dimension. Define a permutation symmetric, homogeneous multilinear polynomial random matrix of degree d as*

$$\mathbf{F}(\mathbf{x}) = \sum_{\mathbf{i}\in\mathcal{T}_n^d}\left(\mathbf{A_i}\cdot\prod_{j\in\{i_1,\dots,i_d\}} x_j\right).$$

Let $a, b, c \in \mathbb{Z}_{\ge 0}$ and $d = a+b+c$. Then for $2 \le t \le \infty$,

$$\mathbb{E}\left\|\mathbf{F}(\mathbf{x})\right\|_{4t}^{4t} \le \sum_{\substack{a,b,c:\\ a+b+c=d}} (48dt)^{4dt}\cdot L^{4ct}\left\|\mathbf{F}_{a,b,c}\right\|_{4t}^{4t}.$$

We remark that the condition that $|x_i| \le L$ can be replaced by a condition on the growth of moments of x_i, which is true for subgaussian variables. The above bound is in terms of the norms of a constant number of *deterministic* matrices $\mathbf{F}_{a,b,c}$, as is also the case for general moment bounds on scalar polynomials [1]. We will see later that these deterministic matrices can easily be interpreted in several cases of interest, to recover the bounds obtained via combinatorial methods.

Gaussian polynomial matrices. While the above bounds are only for multilinear polynomials, they can also be extended to *arbitrary* polynomials of independent Gaussian random variables, using standard techniques to approximate them by multilinear polynomials. However, for the case of Gaussian polynomials, the recent Poincaré inequalities of Huang and Tropp [14] directly yield a simple bound, which is easier to apply. They show that for a Hermitian matrix-valued function $\mathbf{H}$ of Gaussian valued random variables

$$\mathbb{E}\left\|\mathbf{H}(\mathbf{x}) - \mathbb{E}\mathbf{H}(\mathbf{x})\right\|_{2t}^{2t} \le (\sqrt{2}t)^{2t}\cdot\mathbb{E}\,\mathrm{tr}\left(\sum_{i=1}^n(\partial_i\mathbf{H}(\mathbf{x}))^2\right)^t.$$

As before, one can apply this argument inductively to obtain a bound in terms of the matrices $\mathbf{F}_{a,b}$ which are defined similarly as the matrices $\mathbf{F}_{a,b,c}$ before, with $c = 0$. Since we no longer have $\mathbb{E}\mathbf{F}_{a,b} = \mathbf{0}$ for $a+b < d$, the bound is in terms of the expected matrices $\mathbb{E}\mathbf{F}_{a,b}$ for all a, b with $a+b \le d$.

▶ **Theorem 2** (Restatement of Theorem 14). *Let $\mathbf{x} \sim \mathcal{N}(0, \mathbb{I}_n)$ and $\{\mathbf{A_i}\}_{\mathbf{i}\in[n]^d}$ be a sequence of deterministic matrices of the same dimension. Define a degree-d homogeneous Gaussian polynomial random matrix as*

$$\mathbf{F}(\mathbf{x}) = \sum_{\mathbf{i}\in[n]^d} \left(\mathbf{A_i} \cdot \prod_{j\in\{i_1,\ldots,i_d\}} x_j \right).$$

Let $a, b \in \mathbb{Z}_{\geq 0}$. Then for $2 \leq t \leq \infty$,

$$\mathbb{E}\,\|\mathbf{F}(\mathbf{x}) - \mathbb{E}\mathbf{F}(\mathbf{x})\|_{2t}^{2t} \;\leq\; (2^d\sqrt{2}t)^{2t} \left(\sum_{1\leq a+b\leq d} \|\mathbb{E}\mathbf{F}_{a,b}\|_{2t}^{2t} \right).$$

While the above theorem is stated for homogeneous polynomials, the proof is identical also for the non-homogeneous ones, with the sum in the definition being over all tuple sizes.

Do try this at home: applying the framework

We now discuss how to interpret the matrices $\mathbf{F}_{a,b,c}$ arising in the bound in Theorem 1 to recover the norm bounds for graph matrices derived via combinatorial methods. This is discussed with more formal details in Section 6, but we present an intuitive (at least for the authors) version of the argument here.

Graph matrices are defined using constant-sized template "shapes" and provide a convenient basis for expressing (large) random matrices where the the entries are low-degree polynomials in the (normalized) indicators of edges in a $G_{n,p}$ random graph. Such matrices and their norm bounds have been used for a large number of applications [2].

Let $N = \binom{n}{2}$ and fix a canonical bijection between the space $[N]$ and $\binom{[n]}{2}$, and let $[N]$ index the space of all possible edges in a random graph on n vertices. For each $e = \{i, j\} \in \binom{[n]}{2}$, for $\Delta_p = \sqrt{(1-p)/p}$ let x_e be independently $1/\Delta_p$ with probability $(1-p)$ and $-\Delta_p$ with probability p, so that $\mathbb{E}x_e = 0$, $\mathbb{E}x_e^2 = 1$, and $|x_e| \leq \Delta_p$.

Let $(V(\tau), E(\tau))$ be a graph on k vertices with d edges, and let U_τ, V_τ be *ordered* subsets of $V(\tau)$ of size k_1 and k_2 respectively. The tuple $\tau = (V(\tau), E(\tau), U_\tau, V_\tau)$ is referred to as a "shape" and is used to define the following graph matrix $\mathbf{M}_\tau$ with rows and columns indexed by $\mathcal{T}_n^{k_1}$ and $\mathcal{T}_n^{k_2}$ respectively

$$\mathbf{M}_\tau[\mathbf{i}, \mathbf{j}] \;:=\; \sum_{\psi\in\mathcal{T}_n^k} \mathbb{1}_{\{\psi(U_\tau)=\mathbf{i}\}} \cdot \mathbb{1}_{\{\psi(V_\tau)=\mathbf{j}\}} \cdot \prod_{e_0\in E(\tau)} x_{\psi(e_0)}\,,$$

where we interpret a tuple $\psi \in \mathcal{T}_n^k$ as an injective function $\psi : V(\tau) \to [n]$ in the canonical way. If the function ψ maps U_τ to $\mathbf{i}$ (the row-index) and V_τ to $\mathbf{j}$ (the column index), we add the monomial to $\mathbf{M}_\tau[\mathbf{i}, \mathbf{j}]$ corresponding to the product of the images of all edges in the "pattern" $E(\tau)$ under ψ. Note that each nonzero entry of the matrix-valued function $\mathbf{M}$ is a multilinear homogeneous polynomial in the variables $\{x_e\}_{e\in\binom{[n]}{2}}$ with degree $d = |E(\tau)|$. We denote $\mathbf{M}_\tau$ by $\mathbf{F}$ for ease of notation in the discussion below.

As will be apparent from the analysis below, the norm bounds for such matrices behave different characterizations in the "dense" regime where $p = \Omega(1)$ and the "sparse" one where $p = o(1)$. This is because the subgaussian norm $L = \Delta_p \approx p^{-1/2}$ is bounded in the first case, and growing in the second case. The bounds for the first case are stated in terms of the size the minimum vertex separator separating U_τ from V_τ in the shape τ. For the second case, one needs to augment the defintion of minimum vertex separator, to use a cost based on the subgaussian norm Δ_p. We will show how to recover both these characterizations, using the matrices $\mathbf{F}_{a,b,c}$ given by Theorem 1.

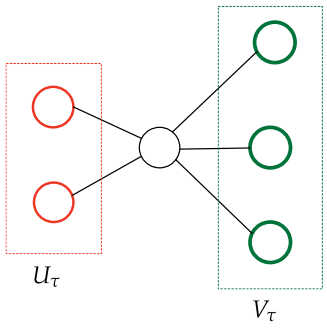


Figure 1 A shape τ.

Dense graph matrices. We start by considering $p = 1/2$ so that the random variables x_e are just independent Rademacher variables. Taking d to be constant, the bound in Theorem 1 involves only a constant number of matrices, and will depend on the dominant term. To understand each of the terms, we first consider the matrices derived after one step of the recursion.

$$\mathbf{F}_{1,0,0} = \begin{bmatrix}\partial_{x_1}\mathbf{F} & \dots & \partial_{x_N}\mathbf{F}\end{bmatrix}^\mathsf{T}, \quad \mathbf{F}_{0,1,0} = \begin{bmatrix}\partial_{x_1}\mathbf{F} & \dots & \partial_{x_N}\mathbf{F}\end{bmatrix} \text{ and } \mathbf{F}_{0,0,1} = \begin{bmatrix}\partial_{x_1}\mathbf{F} & & \\ & \ddots & \\ & & \partial_{x_N}\mathbf{F}\end{bmatrix}.$$

Since $\operatorname{tr}\left(\sum_i \mathbf{A}_i\mathbf{A}_i^\mathsf{T}\right)^{2t} + \operatorname{tr}\left(\sum_i \mathbf{A}_i^\mathsf{T}\mathbf{A}_i\right)^{2t} \geq \sum_i \operatorname{tr}\left(\mathbf{A}_i\mathbf{A}_i^\mathsf{T}\right)^{2t}$ and $L = 1$, we will have that $\|\mathbf{F}_{1,0,0}\|_{4t}^{4t} + \|\mathbf{F}_{0,1,0}\|_{4t}^{4t} \geq L^{4t} \cdot \|\mathbf{F}_{0,0,1}\|_{4t}^{4t}$. This will also be true (up to constant factors) for any constant L, since we interested in the $4t$-th root of the above trace. Using a similar argument, we can ignore terms with $c > 0$ for now (we will come back to these in the sparse case) and focus on matrices $\mathbf{F}_{a,b,0}$.

We now interpret the matrices $\mathbf{F}_{1,0,0}$ and $\mathbf{F}_{0,1,0}$. The row-space of $\mathbf{F}_{1,0,0}$ is now indexed by $(\mathbf{i}, e)$ for $e \in \binom{[n]}{2}$ (respectively $(\mathbf{j}, e)$ for the column space of $\mathbf{F}_{0,1,0}$). We now include terms corresponding to maps ψ for which $\psi(U_\tau) = \mathbf{i}$, $\psi(V_\tau) = \mathbf{j}$ and $\psi(e_0) = e$ for some $e_0 \in E(\tau)$ so that $\partial x_e \mathbf{F}[\mathbf{i}, \mathbf{j}] \neq 0$.

Decomposing $\mathbf{F}_{1,0,0}$ into $|E(\tau)|$ matrices (one for each choice of e_0), we can consider the $\mathbf{F}_{1,0,0,e_0}$ as simply a new graph matrix we have *deleted* the edge e_0 and included both its end points in U_τ (respectively in V_τ for $\mathbf{F}_{0,1,0}$) to obtain a new shape τ'. Similarly, each $\mathbf{F}_{a,b,0}$ can be decomposed into $|E(\tau)|^{a+b}$ graph matrices, where we delete $a+b$ edges in total, and include the endpoints in U_τ for a of them and V_τ for b of them. This is illustrated in Figure 2 where we color the edges red or green depending on their inclusion in U_τ or V_τ.

The bound in Theorem 1 is then in terms of these new graph matrices where we delete all d edges, and split their endpoints between U_τ or V_τ to obtain $U', V' \subseteq V(\tau) = [k]$. For an entry $\mathbf{M}[\mathbf{i}', \mathbf{j}']$ of any such matrix, there is at most one $\psi : V(\tau) \to [n]$ such that $\psi(U') = \mathbf{i}'$ and $\psi(V') = \mathbf{j}'$ (assuming τ has no isolated vertices), and $\mathbf{M}[\mathbf{i}', \mathbf{j}'] = \mathbb{1}_{\{\psi(U')=\mathbf{i}'\}} \cdot \mathbb{1}_{\{\psi(V')=\mathbf{j}'\}}$. Taking $S = U' \cap V'$, this matrix can be simply written as a block-diagonal matrix with $n^{|S|}$ of size $n^{|U'\setminus S|} \times n^{|V'\setminus S|}$. It is easy to check that the spectral norm of such a matrix is $n^{(|U'\setminus S|+|V'\setminus S|)/2} = n^{(k-|S|)/2}$ since $|U' \cup V'| = k$ and $S = U' \cap V'$.

Thus, each of the terms in the bound will correspond to graph matrices with different U' and V' (obtained by coloring edges red or green according to the above process) and the dominant term will be the one with the smallest value of $|U' \cap V'|$. We now claim that

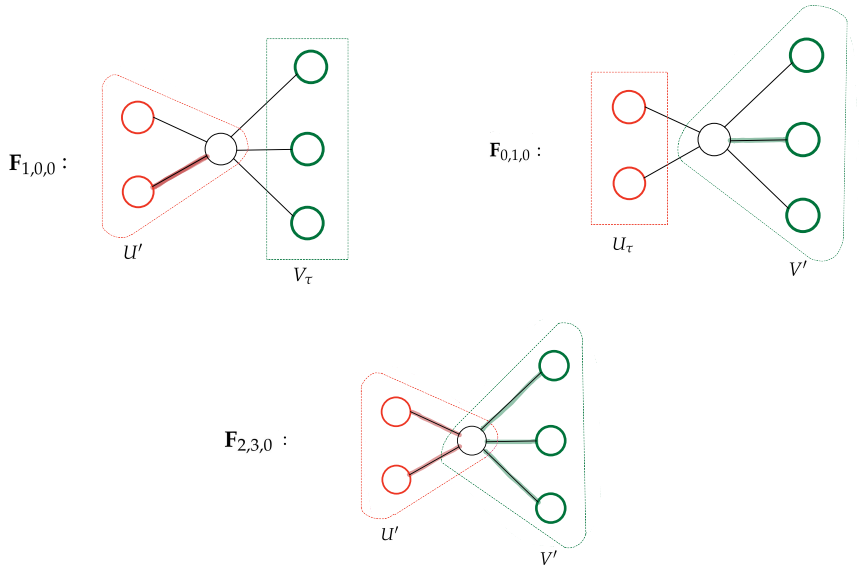


Figure 2 $\mathbf{F}_{1,0,0}$, $\mathbf{F}_{0,1,0}$ and $\mathbf{F}_{2,3,0}$.

$U' \cap V'$ must be a vertex separator separating U_τ and V_τ in the shape τ i.e., any path between them must pass through $U' \cap V'$. If the path contains both red and green edges, then it contains one vertex with both red and green edges incident on it, which is then in $U' \cap V'$ (since U' is obtain by adding endpoint of red edges to U_τ and V' similarly for green edges). If path is entirely red, then we have an endpoint of a red edge in V_τ which is a vertex in $U' \cap V'$, and similarly for green paths.

Thus, we have $n^{(k-|S|)/2} \leq n^{(k-r)/2}$, where r is the size of the minimum vertex separator between U_τ and V_τ, which recovers the known bound for dense graph matrices [2].

Sparse graph matrices. The argument for sparse graph matrices is almost the same as above except now we also need to consider terms of the form $\mathbf{F}_{a,b,c}$ for $c > 0$. Just as we interpreted $\mathbf{F}_{a,b,0}$ as coloring a edges from $E(\tau)$ as red and including in $U' \supseteq U_\tau$ and b green edges in $V' \supseteq V_\tau$, we can now interpret $\mathbf{F}_{a,b,c}$ as having c edges whose endpoints are included in *both* U' and V' (say these edges are colored yellow). This simply reflects the fact that the derivatives in the definition of $\mathbf{F}_{a,b,c+1}$ are placed as diagonal blocks.

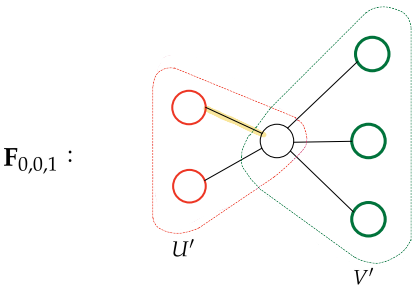


Figure 3 $\mathbf{F}_{0,0,1}$.

As we saw above, increasing the intersection of U' and V' decreases the norm, and so the matrices with $c > 0$ should have a smaller Schatten norm. However, they are now included in the bound with a multiplicative factor of L^c. Thus, we simply look for a vertex separator S maximizing $L^{e(S)} \cdot n^{(k-|S|)/2}$ where $e(S)$ counts the (yellow) edges contained in $S = U' \cap V'$. This is precisely the "sparse vertex separator" which determines the norm bound for such sparse graph matrices [16, 29].

2 Preliminaries and Notation

Vectors are denoted by bold face lower case letters $\mathbf{x}, \mathbf{y}$ and $\mathbf{z}$. Deterministic matrices are denoted by bold face upper case letters $\mathbf{A}$ and $\mathbf{B}$. Matrix-valued functions are denoted by bold face upper case letters $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{F}$, $\mathbf{M}$, $\mathbf{H}$ and $\mathbf{P}$.

Sets and indices. For a set S, let $|S|$ denote the number of distinct elements in S. We denote $[n] = \{1, \ldots, n\}$ for any positive integer n. For $\mathbf{i} := (i_1, \ldots, i_m) \in [n]^m$, define

$$\mathcal{T}_n^m := \{(i_1, \ldots, i_m) : \forall j, k \in [m],\ j \neq k \Rightarrow i_j \neq i_k\}.$$

Matrix norms. Let $\mathbb{R}^{d_1 \times d_2}$ be the space of all $d_1 \times d_2$ real matrices. Let $\mathbb{H}^d$ be the subspace of $\mathbb{R}^{d \times d}$ containing all Hermitian matrices. We write $\|\cdot\|_2$ for the ℓ_2 operator norm, $\|\cdot\|_F$ for the Frobenius norm, and $\mathrm{tr}(\cdot)$ for the trace.

▶ **Definition 3** (Schatten norm). *For $\mathbf{A} \in \mathbb{R}^{d_1 \times d_2}$ and $t \geq 1$, the Schatten $2t$-norm is defined as*

$$\|\mathbf{A}\|_{2t} := (\mathrm{tr}(\mathbf{A}^\mathsf{T}\mathbf{A})^t)^{1/2t}.$$

Polynomial random matrices. Let $(\Omega, \mathcal{F}, \mu)$ be a probability space. Introduce the random vector $\mathbf{x} := (x_1, \ldots, x_n) \in \Omega$ where $x_1, \ldots, x_n$ are independent random variables. A polynomial random matrix $\mathbf{F}(\mathbf{x})$ is a matrix-valued polynomial $\mathbf{F} : \Omega \to \mathbb{R}^{d_1 \times d_2}$. We can construct a polynomial random matrix $\mathbf{F}(\mathbf{x})$ by drawing a copy of $\mathbf{x} \sim \mu$. Note that despite $\mathbf{x}$ having independent entries, the entries of $\mathbf{F}(\mathbf{x})$ can be dependent (or correlated) with each other. Unlike Wigner matrices whose entries are i.i.d., the polynomial random matrices can have dependencies among its entries.

▶ **Definition 4** (Permutation Symmetric Property). *Suppose $\mathbf{F}(\mathbf{x})$ is a degree-d homogeneous multilinear polynomial random matrix, i.e.,*

$$\mathbf{F}(\mathbf{x}) = \sum_{\mathbf{i} \in \mathcal{T}_n^d} \left(\mathbf{A}_\mathbf{i} \cdot \prod_{j \in \{i_1, \ldots, i_d\}} x_j \right),$$

where $\{\mathbf{A}_\mathbf{i}\}_{\mathbf{i} \in \mathcal{T}_n^d}$ is a multi-indexed sequence of deterministic matrices of the same dimension. $\mathbf{F}(\mathbf{x})$ is permutation symmetric if $\mathbf{A}_{i_1, \ldots, i_d} = \mathbf{A}_{i_{\sigma(1)}, \ldots, i_{\sigma(d)}}$ for any permutation $\sigma \in \mathfrak{S}_d$ and any $(i_1, \ldots, i_d) \in \mathcal{T}_n^d$.

Partial derivatives. Let $\mathbf{F}(\mathbf{x})$ be an n-variate polynomial random matrix of degree D. For $a, b, c \in \mathbb{Z}_{\geq 0}$ and $d = a + b + c$, we consider the block matrices $\mathbf{F}_{a,b,c}$ containing all d-th order partial derivatives of $\mathbf{F}(\mathbf{x})$. We define $\mathbf{F}_{a,b,c}$ recursively as $\mathbf{F}_{0,0,0} = \mathbf{F}(\mathbf{x})$ and

$$\mathbf{F}_{a+1,b,c} = \begin{bmatrix} \partial_{x_1} \mathbf{F}_{a,b,c} & \ldots & \partial_{x_n} \mathbf{F}_{a,b,c} \end{bmatrix}^\mathsf{T},$$

$$\mathbf{F}_{a,b+1,c} = \begin{bmatrix} \partial_{x_1}\mathbf{F}_{a,b,c} & \dots & \partial_{x_n}\mathbf{F}_{a,b,c} \end{bmatrix},$$

$$\mathbf{F}_{a,b,c+1} = \begin{bmatrix} \partial_{x_1}\mathbf{F}_{a,b,c} & & \\ & \ddots & \\ & & \partial_{x_n}\mathbf{F}_{a,b,c} \end{bmatrix}.$$

It is evident from the definition that the block matrices $\mathbf{F}_{a+1,b,c}$, $\mathbf{F}_{a,b+1,c}$, $\mathbf{F}_{a,b,c+1}$ are assembled from sub-blocks $\{\partial_{x_i}\mathbf{F}_{a,b,c}\}_{i=1}^n$ arranged vertically, horizontally and diagonally respectively. The order of increment between a and b doesn't affect the resulting matrix, i.e.,

$$\mathbf{F}_{a+1,b+1,c} = \begin{bmatrix} \partial_{x_1}\mathbf{F}_{a,b+1,c} \\ \vdots \\ \partial_{x_n}\mathbf{F}_{a,b+1,c} \end{bmatrix} = \begin{bmatrix} \partial_{x_1}\mathbf{F}_{a+1,b,c} & \dots & \partial_{x_n}\mathbf{F}_{a+1,b,c} \end{bmatrix} = \begin{bmatrix} \partial_{x_1}\partial_{x_1}\mathbf{F}_{a,b,c} & \dots & \partial_{x_1}\partial_{x_n}\mathbf{F}_{a,b,c} \\ & \vdots & \\ \partial_{x_n}\partial_{x_1}\mathbf{F}_{a,b,c} & \dots & \partial_{x_n}\partial_{x_n}\mathbf{F}_{a,b,c} \end{bmatrix}.$$

However, the order of increment involving c affects the resulting matrix, i.e.,

$$\begin{bmatrix} \partial_{x_1}\mathbf{F}_{a,b,c+1} & \dots & \partial_{x_n}\mathbf{F}_{a,b,c+1} \end{bmatrix} \neq \begin{bmatrix} \partial_{x_1}\mathbf{F}_{a,b+1,c} & & \\ & \ddots & \\ & & \partial_{x_n}\mathbf{F}_{a,b+1,c} \end{bmatrix}.$$

Nevertheless, the order of increment doesn't affect the Schatten norm. Since we will only be interested in the Schatten norms of these matrices, we will simply label them as $\mathbf{F}_{a,b+1,c+1}$ without specifying the order. It is easy to see that $\mathbf{F}_{a,b,c}$ is a deterministic matrix if $a+b+c = D$ and $\mathbf{F}_{a,b,c} = \mathbf{0}$ if $a+b+c > D$.

Constants. We denote C for some universal constants and C_a for constants depending only on some parameter a. The values of the constants may differ from one instance to another.

3 Decoupling Inequalities

In this work, we focus on decoupling inequalities for moments, which is fairly elementary and can be interpreted combinatorially as partitioning the random vector $\mathbf{x}$.

▶ **Lemma 5** (see [30] Lemma 6.21). *Let $\mathbf{x} = \{x_j\}_{j=1}^n$ be a sequence of independent random variables with $\mathbb{E}x_j = 0$ for all $j = 1, \dots, n$. Let $\{\mathbf{B}_\mathbf{i}\}_{\mathbf{i}\in\mathcal{T}_n^d}$ be a multi-indexed sequence of deterministic matrices of the same dimension. Then for $1 \le t \le \infty$*

$$\mathbb{E}\left\| \sum_{\mathbf{i}\in\mathcal{T}_n^d} \mathbf{B}_\mathbf{i} \cdot x_{i_1} \dots x_{i_d} \right\|_{2t} \le d^d \cdot \mathbb{E}\left\| \sum_{\mathbf{i}\in\mathcal{T}_n^d} \mathbf{B}_\mathbf{i} \cdot x_{i_1}^{(1)} \dots x_{i_d}^{(d)} \right\|_{2t},$$

where $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(d)}$ denote d independent copies of $\mathbf{x}$.

Improved Decoupling for Random Variables with Structured Indices

For a generic degree d multilinear polynomial random matrices, an application of Lemma 5 yields a decoupling constant of d^d, which might not be optimal for polynomial random matrices with additional structures. We will show that the additional structures enable us to give a tighter decoupling inequality through a careful partitioning.

Building on Lemma 5, we prove a decoupling inequality for polynomial random matrices, in which the indices of random variables have a graph structure. Let $G = (V, E)$ be a fixed simple graph with $|V| = k$, and $|E| = d$. We denote the vertices by $V(G) = \{v_1, \ldots, v_k\}$ and the ordered edge set by $E(G) = \{(i_1, j_1), \ldots, (i_d, j_d)\}$. For any $\varphi \in \mathcal{T}_n^k$, we write $\varphi(i)$ for the i-th component of φ and $\varphi(E)$ for $\{(\varphi(i_1), \varphi(j_1)), \ldots, (\varphi(i_d), \varphi(j_d))\}$.

▶ **Lemma 6.** *Let $\mathbf{z} = \{Z_{\varphi(i),\varphi(j)}\}_{\varphi \in \mathcal{T}_n^k, (i,j) \in E}$ be a double sequence of independent random variables with $\mathbb{E} Z_{\varphi(i),\varphi(j)} = 0$ for all $(i, j) \in E$ and $\varphi \in \mathcal{T}_n^k$. Let $\{\mathbf{B}_{\varphi(E)}\}_{\varphi \in \mathcal{T}_n^k}$ be a multi-indexed sequence of deterministic matrices of the same dimension. Then for $d \le k(k-1)$ and $1 \le t \le \infty$,*

$$\mathbb{E} \left\| \sum_{\varphi \in \mathcal{T}_n^k} \mathbf{B}_{\varphi(E)} \cdot \prod_{(i,j) \in E} Z_{\varphi(i),\varphi(j)} \right\|_{2t} \le k^k \cdot \mathbb{E} \left\| \sum_{\varphi \in \mathcal{T}_n^k} \mathbf{B}_{\varphi(E)} \cdot Z^{(1)}_{\varphi(i_1),\varphi(j_1)} \cdots Z^{(d)}_{\varphi(i_d),\varphi(j_d)} \right\|_{2t},$$

where $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(d)}$ denote d independent copies of $\mathbf{z}$.

▶ Remark 7. It is crucial for the indices of all monomials to share the same structure so that d independent copies of $\mathbf{z}$ suffices. Otherwise, $k(k-1)$ independent copies are needed in general.

Proof. Consider a random partition of the ground set $[n]$ into k parts and let $\mathbf{r}$ be the *vertex partitioner*. Formally, let $\mathbf{r} = (r_1, \ldots, r_n)$ be a sequence of independent random variables with each r_p uniformly distributed on $\{1, \ldots, k\}$, i.e., for $1 \le p \le n$,

$$\mathbb{P}(r_p = 1) = \mathbb{P}(r_p = 2) = \cdots = \mathbb{P}(r_p = k) = 1/k.$$

Define an event $A := \{r_{\varphi(v_1)} = 1, \cdots, r_{\varphi(v_k)} = k\}$ for each $\varphi \in \mathcal{T}_n^k$. Conditioned on $\mathbf{z}$,

$$\mathbb{E}_{\mathbf{r}} \left[\mathbb{1}_A(r_{\varphi(v_1)}, \ldots, r_{\varphi(v_k)}) \right] = 1/k^k.$$

It follows that

$$\begin{aligned} F :&= \mathbb{E}_{\mathbf{z}} \left\| \sum_{\varphi \in \mathcal{T}_n^k} \mathbf{B}_{\varphi(E)} \cdot \prod_{(i,j) \in E} Z_{\varphi(i),\varphi(j)} \right\|_{2t} \\ &= k^k \cdot \mathbb{E}_{\mathbf{z}} \left\| \mathbb{E}_{\mathbf{r}} \left[\sum_{\varphi \in \mathcal{T}_n^k} \mathbb{1}_A(r_{\varphi(v_1)}, \ldots, r_{\varphi(v_k)}) \cdot \mathbf{B}_{\varphi(E)} \cdot \prod_{(i,j) \in E} Z_{\varphi(i),\varphi(j)} \right] \right\|_{2t} \\ &\le k^k \cdot \mathbb{E}_{\mathbf{r}} \mathbb{E}_{\mathbf{z}} \left\| \sum_{\varphi \in \mathcal{T}_n^k} \mathbb{1}_A(r_{\varphi(v_1)}, \ldots, r_{\varphi(v_k)}) \cdot \mathbf{B}_{\varphi(E)} \cdot \prod_{(i,j) \in E} Z_{\varphi(i),\varphi(j)} \right\|_{2t}, \end{aligned}$$

where the last step is due to Jensen's inequality and Fubini's theorem. Since the expectation over $\mathbf{r}$ satisfies the inequality, this implies the existence of an $\mathbf{r}^*$ satisfying the inequality.

Fix $\mathbf{r}^*$ and define the *vertex partitioning* with respect to $\mathbf{r}^*$ as

$$P_1 = \{p \in [n] : r_p^* = 1\}, \ \ldots \ , P_k = \{p \in [n] : r_p^* = k\},$$

which in turn induces an *edge partitioning*,

$$P_e \ := \ \{(P_i, P_j)\}_{(i,j) \in \mathcal{T}_k^2}.$$

Notice that there are $k(k-1)$ elements in P_e, but we only have d edges to partition. So we select d elements out of P_e in the following way. Fix an arbitrary $\varphi^* \in \mathcal{T}_n^k$, choose

$$P_1^* = (P_{\varphi^*(i_1)}, P_{\varphi^*(j_1)}), \ \ldots \ , P_d^* = (P_{\varphi^*(i_d)}, P_{\varphi^*(j_d)}).$$

Let's call $\{P_1^*, \dots P_d^*\}$ a *d-edge partitioning*. In other words, we may select any d elements out of P_e to form a *d-edge partitioning* so long as the pattern of the indices of (P_i, P_j)'s matches that of $E(G)$. Since the elements in P_e have no intersections, the partitions in $\{P_1^*, \dots, P_d^*\}$ also have no intersections. Since $Z_{\varphi(i),\varphi(j)}$'s are independent random variables, replacing $Z_{\varphi(i_1),\varphi(j_1)} \cdots Z_{\varphi(i_d),\varphi(j_d)}$ with $Z^{(1)}_{\varphi(i_1),\varphi(j_1)} \cdots Z^{(d)}_{\varphi(i_d),\varphi(j_d)}$ for any $\varphi(E) \in P_1^* \times \cdots \times P_d^*$ will not change the distribution. Hence

$$F \le k^k \cdot \mathbb{E} \left\| \sum_{\varphi(E) \in P_1^* \times \cdots \times P_d^*} \mathbf{B}_{\varphi(E)} \cdot Z^{(1)}_{\varphi(i_1),\varphi(j_1)} \cdots Z^{(d)}_{\varphi(i_d),\varphi(j_d)} \right\|_{2t}. \tag{1}$$

Denote all variables in $P_1^* \times \cdots \times P_d^*$ by

$$\mathcal{Z} := \left\{ Z^{(1)}_{\varphi(i_1),\varphi(j_1)}, \ \dots, \ Z^{(d)}_{\varphi(i_d),\varphi(j_d)} : \ \forall \ (\varphi(i_1),\varphi(j_1)) \in P_1^*, \ \dots \ , (\varphi(i_d),\varphi(j_d)) \in P_d^* \right\},$$

and denote the rest of the variables by $\mathcal{Z}^c$. It remains to show that the sum on the right-hand side of (1) is over all $\varphi \in \mathcal{T}_n^k$. Observe that

$$\mathbb{E}_{\mathcal{Z}^c} \left[\sum_{\varphi(E) \notin P_1^* \times \cdots \times P_d^*} \mathbf{B}_{\varphi(E)} \cdot Z^{(1)}_{\varphi(i_1),\varphi(j_1)} \cdots Z^{(d)}_{\varphi(i_d),\varphi(j_d)} \ \middle| \ \mathcal{Z} \right] = \mathbf{0}. \tag{2}$$

Denote $P_1^* \times \cdots \times P_d^*$ by $\mathcal{P}$. Since the sum in (2) has conditional expectation zero, we can add it to (1) to get

$$F \le k^k \cdot \mathbb{E}_{\mathcal{Z}} \left\| \sum_{\varphi(E) \in \mathcal{P}} \mathbf{B}_{\varphi(E)} \cdot Z^{(1)}_{\varphi(i_1),\varphi(j_1)} \cdots Z^{(d)}_{\varphi(i_d),\varphi(j_d)} + \mathbb{E}_{\mathcal{Z}^c} \left[\sum_{\varphi(E) \notin \mathcal{P}} \mathbf{B}_{\varphi(E)} \cdot Z^{(1)}_{\varphi(i_1),\varphi(j_1)} \cdots Z^{(d)}_{\varphi(i_d),\varphi(j_d)} \ \middle| \ \mathcal{Z} \right] \right\|_{2t}.$$

Since the two sums on the right-hand side are independent conditioned on $\mathcal{Z}$, conditional application of Jensen's inequality yields the desired inequality,

$$F \le k^k \cdot \mathbb{E}_{\mathbf{z}^{(1)},\dots,\mathbf{z}^{(d)}} \left\| \sum_{\varphi \in \mathcal{T}_n^k} \mathbf{B}_{\varphi(E)} \cdot Z^{(1)}_{\varphi(i_1),\varphi(j_1)} \cdots Z^{(d)}_{\varphi(i_d),\varphi(j_d)} \right\|_{2t}. \qquad \blacktriangleleft$$

4 Multilinear Polynomial Random Matrices

In this section, we prove a moment bound for multilinear polynomial random matrices with bounded, normalized random variables. We follow our recursion framework by decoupling the polynomial random matrices first and then apply the matrix Rosenthal inequality recursively to sums of (conditionally) independent random matrices.

To this end, we derive the non-Hermitian matrix Rosenthal inequality from the Hermitian matrix Rosenthal inequality by Hermitian dilation (see [34] Sec. 2.1.16).

▶ **Lemma 8** (Non-Hermitian Matrix Rosenthal Inequality). *Suppose that $t = 1$ or $t \ge 1.5$. Consider a finite sequence $\{\mathbf{Y}_k\}_{k \ge 1}$ of centered, independent, random matrices, and assume that $\mathbb{E} \|\mathbf{Y}_k\|_{4t}^{4t} < \infty$. Then*

$$\mathbb{E} \left\| \sum_k \mathbf{Y}_k \right\|_{4t}^{4t} \le (16t)^{3t} \cdot \left\{ \left\| \left(\sum_k \mathbb{E} \mathbf{Y}_k \mathbf{Y}_k^{\mathsf{T}} \right)^{1/2} \right\|_{4t}^{4t} + \left\| \left(\sum_k \mathbb{E} \mathbf{Y}_k^{\mathsf{T}} \mathbf{Y}_k \right)^{1/2} \right\|_{4t}^{4t} \right\} + (8t)^{4t} \cdot \left(\sum_k \mathbb{E} \|\mathbf{Y}_k\|_{4t}^{4t} \right).$$

Proof. For a sequence of Hermitian matrices $\{\mathbf{X}_k\}_{k\geq 1}$ satisfying all the assumptions above, we have (Hermitian) matrix Rosenthal inequality [22] Corollary 7.4,

$$\mathbb{E}\left\|\sum_k \mathbf{X}_k\right\|_{4t}^{4t} \leq (16t)^{3t}\cdot\left\|\left(\sum_k \mathbb{E}\mathbf{X}_k^2\right)^{1/2}\right\|_{4t}^{4t} + (8t)^{4t}\cdot\sum_k \mathbb{E}\left\|\mathbf{X}_k\right\|_{4t}^{4t}.$$

We use Hermitian dilation to extend the inequality to non-Hermitian matrices by setting $\mathbf{X}_k = \mathcal{H}(\mathbf{Y}_k)$ and notice that

$$\mathbb{E}\left\|\sum_k \begin{bmatrix}\mathbf{0} & \mathbf{Y}_k\\ \mathbf{Y}_k^\intercal & \mathbf{0}\end{bmatrix}\right\|_{4t}^{4t} = \mathbb{E}\,\mathrm{tr}\left(\begin{bmatrix}((\sum_k \mathbf{Y}_k)(\sum_k \mathbf{Y}_k^\intercal))^{2t} & \mathbf{0}\\ \mathbf{0} & ((\sum_k \mathbf{Y}_k^\intercal)(\sum_k \mathbf{Y}_k))^{2t}\end{bmatrix}\right) = 2\mathbb{E}\left\|\sum_k \mathbf{Y}_k\right\|_{4t}^{4t},$$

$$\left\|\left(\sum_k \mathbb{E}\begin{bmatrix}\mathbf{Y}_k\mathbf{Y}_k^\intercal & \mathbf{0}\\ \mathbf{0} & \mathbf{Y}_k^\intercal\mathbf{Y}_k\end{bmatrix}\right)^{1/2}\right\|_{4t}^{4t} = \mathrm{tr}\left(\begin{bmatrix}\sum_k \mathbb{E}\mathbf{Y}_k\mathbf{Y}_k^\intercal & \mathbf{0}\\ \mathbf{0} & \sum_k \mathbb{E}\mathbf{Y}_k^\intercal\mathbf{Y}_k\end{bmatrix}\right)^{2t}$$

$$= \left\|\left(\sum_k \mathbb{E}\mathbf{Y}_k\mathbf{Y}_k^\intercal\right)^{1/2}\right\|_{4t}^{4t} + \left\|\left(\sum_k \mathbb{E}\mathbf{Y}_k^\intercal\mathbf{Y}_k\right)^{1/2}\right\|_{4t}^{4t},$$

$$\sum_k \mathbb{E}\left\|\begin{bmatrix}\mathbf{0} & \mathbf{Y}_k\\ \mathbf{Y}_k^\intercal & \mathbf{0}\end{bmatrix}\right\|_{4t}^{4t} = 2\sum_k \mathbb{E}\left\|\mathbf{Y}_k\right\|_{4t}^{4t}.$$

Hence, the result follows. ◀

▷ Claim 9. Let $\mathbf{F}(\mathbf{x})$ be a homogeneous multilinear polynomial random matrix of degree D. Let $\mathbf{F}(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(D)})$ be the decoupled $\mathbf{F}(\mathbf{x})$, i.e.,

$$\mathbf{F}(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(D)}) = \sum_{\mathbf{j}\in\mathcal{T}_n^D} \mathbf{A}_{\mathbf{j}}\cdot x_{j_1}^{(1)}\cdots x_{j_D}^{(D)},$$

where $\{\mathbf{A}_{\mathbf{k}}\}_{\mathbf{k}\in\mathcal{T}_n^D}$ is a multi-indexed sequence of deterministic matrices of the same dimension. For some fixed $a,b,c\in\mathbb{Z}_{\geq 0}$ and $k=a+b+c<D$, let $\mathbf{F}_{a,b,c}$ be the block matrix of k-th order partial derivatives of $\mathbf{F}(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(D)})$. Then $\mathbf{F}_{a,b,c}$ is a homogeneous multilinear polynomial random matrix of degree $d=D-k$. Furthermore,

$$\mathbb{E}\left\|\mathbf{F}_{a,b,c}\right\|_{4t}^{4t} \leq (16t)^{3t}\cdot\left(\mathbb{E}\left\|\mathbf{F}_{a+1,b,c}\right\|_{4t}^{4t} + \mathbb{E}\left\|\mathbf{F}_{a,b+1,c}\right\|_{4t}^{4t}\right) + (8t)^{4t}\cdot\mathbb{E}\left\|\mathbf{F}_{a,b,c+1}\right\|_{4t}^{4t}.$$

Proof. Without loss of generality, we differentiate $\mathbf{F}(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(D)})$ with respect to $\mathbf{x}^{(D)}$ first, then $\mathbf{x}^{(D-1)}$, and so on. It is straightforward to see that after $k=a+b+c$ rounds of differentiation, $\mathbf{F}_{a,b,c}(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(d)})$ is a homogeneous polynomial random matrix of degree $d=D-k$. And

$$\mathbf{F}_{a,b,c}(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(d)}) = \sum_{\mathbf{i}\in\mathcal{T}_n^d} \mathbf{B}_{\mathbf{i}}\cdot x_{i_1}^{(1)}\cdots x_{i_d}^{(d)},$$

where $\{\mathbf{B}_{\mathbf{i}}\}_{\mathbf{i}\in\mathcal{T}_n^d}$ is a multi-indexed sequence of deterministic matrices of the same dimension. More specifically, each $\mathbf{B}_{\mathbf{i}}$ is a block matrix whose blocks consist of $\{\mathbf{A}_{\mathbf{j}}\}_{\mathbf{j}\in\mathcal{T}_n^D}$ such that $j_1=i_1,\ldots,j_d=i_d$.

Conditioned on $\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(d-1)}$, $\mathbf{F}_{a,b,c}(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(d)})$ is a sum of centered, (conditionally) independent random matrices in $\mathbf{x}^{(d)}$. It follows that

$$
\begin{aligned}
&\mathbb{E}\left(\left\|\mathbf{F}_{a,b,c}(\mathbf{x}^{(1)},\dots,\mathbf{x}^{(d)})\right\|_{4t}^{4t}\;\middle|\;\mathbf{x}^{(1)},\dots\mathbf{x}^{(d-1)}\right)\\
&=\mathbb{E}\left(\mathbb{E}_{\mathbf{x}^{(d)}}\left(\left\|\sum_{\mathbf{i}\in\mathcal{T}_n^d}\mathbf{B}_{\mathbf{i}}\cdot x_{i_1}^{(1)}\cdots x_{i_d}^{(d)}\right\|_{4t}^{4t}\;\middle|\;\mathbf{x}^{(1)},\dots\mathbf{x}^{(d-1)}\right)\right)\\
&\le(16t)^{3t}\cdot\mathbb{E}\left\|\left(\sum_{i_d=1}^{n}\left(\sum_{\mathbf{i}\in\mathcal{T}_n^d}\mathbf{B}_{\mathbf{i}}\cdot x_{i_1}^{(1)}\cdots x_{i_{d-1}}^{(d-1)}\right)\left(\sum_{\mathbf{i}\in\mathcal{T}_n^d}\mathbf{B}_{\mathbf{i}}\cdot x_{i_1}^{(1)}\cdots x_{i_{d-1}}^{(d-1)}\right)^{\mathsf{T}}\right)^{1/2}\right\|_{4t}^{4t}\\
&\quad+(16t)^{3t}\cdot\mathbb{E}\left\|\left(\sum_{i_d=1}^{n}\left(\sum_{\mathbf{i}\in\mathcal{T}_n^d}\mathbf{B}_{\mathbf{i}}\cdot x_{i_1}^{(1)}\cdots x_{i_{d-1}}^{(d-1)}\right)^{\mathsf{T}}\left(\sum_{\mathbf{i}\in\mathcal{T}_n^d}\mathbf{B}_{\mathbf{i}}\cdot x_{i_1}^{(1)}\cdots x_{i_{d-1}}^{(d-1)}\right)\right)^{1/2}\right\|_{4t}^{4t}\\
&\quad+(8t)^{4t}L^{4t}\cdot\sum_{i_d=1}^{n}\mathbb{E}\left\|\sum_{\mathbf{i}\in\mathcal{T}_n^d}\mathbf{B}_{\mathbf{i}}\cdot x_{i_1}^{(1)}\cdots x_{i_{d-1}}^{(d-1)}\right\|_{4t}^{4t},
\end{aligned}
$$

where the inequality is due to matrix Rosenthal inequality (Lemma 8). Now we have

$$
\begin{aligned}
&\mathbb{E}\left\|\left(\sum_{i_d=1}^{n}\left(\sum_{\mathbf{i}\in\mathcal{T}_n^d}\mathbf{B}_{\mathbf{i}}\cdot x_{i_1}^{(1)}\cdots x_{i_{d-1}}^{(d-1)}\right)\left(\sum_{\mathbf{i}\in\mathcal{T}_n^d}\mathbf{B}_{\mathbf{i}}\cdot x_{i_1}^{(1)}\cdots x_{i_{d-1}}^{(d-1)}\right)^{\mathsf{T}}\right)^{1/2}\right\|_{4t}^{4t}\\
&=\mathbb{E}\left\|\left(\begin{bmatrix}\partial_{x_1^{(d)}}\mathbf{F}_{a,b,c} & \dots & \partial_{x_n^{(d)}}\mathbf{F}_{a,b,c}\end{bmatrix}\begin{bmatrix}\partial_{x_1^{(d)}}\mathbf{F}_{a,b,c}^{\mathsf{T}}\\ \vdots\\ \partial_{x_n^{(d)}}\mathbf{F}_{a,b,c}^{\mathsf{T}}\end{bmatrix}\right)^{1/2}\right\|_{4t}^{4t}=\mathbb{E}\left\|\mathbf{F}_{a,b+1,c}\right\|_{4t}^{4t},
\end{aligned}
$$

$$
\begin{aligned}
&\mathbb{E}\left\|\left(\sum_{i_d=1}^{n}\left(\sum_{\mathbf{i}\in\mathcal{T}_n^d}\mathbf{B}_{\mathbf{i}}\cdot x_{i_1}^{(1)}\cdots x_{i_{d-1}}^{(d-1)}\right)^{\mathsf{T}}\left(\sum_{\mathbf{i}\in\mathcal{T}_n^d}\mathbf{B}_{\mathbf{i}}\cdot x_{i_1}^{(1)}\cdots x_{i_{d-1}}^{(d-1)}\right)\right)^{1/2}\right\|_{4t}^{4t}\\
&=\mathbb{E}\left\|\left(\begin{bmatrix}\partial_{x_1^{(d)}}\mathbf{F}_{a,b,c}^{\mathsf{T}} & \dots & \partial_{x_n^{(d)}}\mathbf{F}_{a,b,c}^{\mathsf{T}}\end{bmatrix}\begin{bmatrix}\partial_{x_1^{(d)}}\mathbf{F}_{a,b,c}\\ \vdots\\ \partial_{x_n^{(d)}}\mathbf{F}_{a,b,c}\end{bmatrix}\right)^{1/2}\right\|_{4t}^{4t}=\mathbb{E}\left\|\mathbf{F}_{a+1,b,c}\right\|_{4t}^{4t},
\end{aligned}
$$

$$
\begin{aligned}
&\sum_{i_d=1}^{n}\mathbb{E}\left\|\sum_{\mathbf{i}\in\mathcal{T}_n^d}\mathbf{B}_{\mathbf{i}}\cdot x_{i_1}^{(1)}\cdots x_{i_{d-1}}^{(d-1)}\right\|_{4t}^{4t}\\
&=\mathbb{E}\,\mathrm{tr}\begin{bmatrix}\partial_{x_1^{(d)}}\mathbf{F}_{a,b,c}\cdot\partial_{x_1^{(d)}}\mathbf{F}_{a,b,c}^{\mathsf{T}} & & \\ & \ddots & \\ & & \partial_{x_n^{(d)}}\mathbf{F}_{a,b,c}\cdot\partial_{x_n^{(d)}}\mathbf{F}_{a,b,c}^{\mathsf{T}}\end{bmatrix}^{2t}=\mathbb{E}\left\|\mathbf{F}_{a,b,c+1}\right\|_{4t}^{4t}.
\end{aligned}
$$

◁

We present a moment bound for permutation symmetric, homogeneous multilinear polynomial random matrices of degree d.

▶ **Theorem 10** (Homogeneous Multilinear Recursion). *Let $\mathbf{x} = \{x_i\}_{i=1}^n$ be a sequences of i.i.d random variables with $\mathbb{E}x_i = 0$, $\mathbb{E}x_i^2 = 1$ and $|x_i| \le L$ for all $1 \le i \le n$. Let $\{\mathbf{A_i}\}_{\mathbf{i}\in\mathcal{T}_n^d}$ be a multi-indexed sequence of deterministic matrices of the same dimension. Define a permutation symmetric, homogeneous multilinear polynomial random matrix of degree d as*

$$\mathbf{F}(\mathbf{x}) = \sum_{\mathbf{i}\in\mathcal{T}_n^d} \left(\mathbf{A_i} \cdot \prod_{j\in\{i_1,\ldots,i_d\}} x_j \right).$$

Let $a, b, c \in \mathbb{Z}_{\ge 0}$ and $d = a + b + c$. Then for $2 \le t \le \infty$,

$$\mathbb{E}\left\|\mathbf{F}(\mathbf{x}) - \mathbb{E}\mathbf{F}(\mathbf{x})\right\|_{4t}^{4t} \le \sum_{\substack{a,b,c:\\ a+b+c=d}} (48dt)^{4dt} \cdot L^{4ct} \left\|\mathbf{F}_{a,b,c}\right\|_{4t}^{4t}.$$

Proof. Let $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(d)}$ be d independent copies of $\mathbf{x}$. We decouple $\mathbf{F}(\mathbf{x})$ by Lemma 5,

$$E := \mathbb{E}\left\|\mathbf{F}((\mathbf{x})\right\|_{4t}^{4t} \le d^{4dt} \cdot \mathbb{E}\left\|\mathbf{F}(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(d)})\right\|_{4t}^{4t}.$$

We take the partial derivative with respect to $\mathbf{x}^{(d)}$ by applying Lemma 8 to get,

$$E \le d^{4dt}(16t)^{3t} \cdot \left(\mathbb{E}\left\|\mathbf{F}_{1,0,0}\right\|_{4t}^{4t} + \mathbb{E}\left\|\mathbf{F}_{0,1,0}\right\|_{4t}^{4t}\right) + d^{4dt}(8t)^{4t}L^{4t} \cdot \mathbb{E}\left\|\mathbf{F}_{0,0,1}\right\|_{4t}^{4t}.$$

Note that $\mathbf{F}_{1,0,0}$, $\mathbf{F}_{0,1,0}$, and $\mathbf{F}_{0,0,1}$ are functions in variables $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(d-1)}$. We take the partial derivative with respect to the rest of the variables until $\mathbf{F}_{a,b,c}$'s become deterministic matrices. Apply Claim 9 recursively and use the permutation symmetric property of $\mathbf{F}(\mathbf{x})$, we have

$$E \le \sum_{\substack{a,b,c:\\ a+b+c=d}} (16dt)^{4dt} \cdot L^{4ct} \frac{d!}{a!b!c!} \left\|\mathbf{F}_{a,b,c}\right\|_{4t}^{4t}.$$

Since $\frac{d!}{a!b!c!} \le \frac{d!}{(d/3)!(d/3)!(d/3)!}$ and $\frac{d!}{(d/3)!(d/3)!(d/3)!} \sim \frac{3\sqrt{3}}{2\pi d} \cdot 3^d$,

$$E \le \sum_{\substack{a,b,c:\\ a+b+c=d}} (48dt)^{4dt} \cdot L^{4ct} \left\|\mathbf{F}_{a,b,c}\right\|_{4t}^{4t}. \qquad \blacktriangleleft$$

As a corollary, we derive a moment bound for general multilinear polynomial random matrices of degree D. The polynomials are split into homogeneous parts and each parts are bounded separately. Let $\mathbf{F}^{=d}(\mathbf{x})$ denote the degree-d homogeneous part of $\mathbf{F}(\mathbf{x})$.

▶ **Corollary 11** (Multilinear Recursion). *Let $\mathbf{x} = \{x_i\}_{i=1}^n$ be a sequences of i.i.d random variables with $\mathbb{E}x_i = 0$, $\mathbb{E}x_i^2 = 1$ and $|x_i| \le L$ for all $1 \le i \le n$. Let $\{\mathbf{A_i}\}_{\mathbf{i}\in\mathcal{T}_n^d}$ be multi-indexed sequences of deterministic matrices of the same dimension. Define a degree-D multilinear polynomial random matrix as*

$$\mathbf{F}(\mathbf{x}) = \sum_{d=1}^{D} \sum_{\mathbf{i}\in\mathcal{T}_n^d} \left(\mathbf{A_i} \cdot \prod_{j\in\{i_1,\ldots,i_d\}} x_j \right).$$

Suppose $\mathbf{F}^{=d}(\mathbf{x})$ *is permutation symmetric for* $1 \le d \le D$. *Let* $a, b, c \in \mathbb{Z}_{\ge 0}$ *and* $d = a+b+c$. *Then for* $2 \le t \le \infty$,

$$\mathbb{E}\,\|\mathbf{F}(\mathbf{x}) - \mathbb{E}\mathbf{F}(\mathbf{x})\|_{4t}^{4t} \;\le\; \sum_{d=1}^{D} D^{4t}(48dt)^{4dt} \left(\sum_{\substack{a,b,c:\\ a+b+c=d}} L^{4ct} \left\|\mathbf{F}_{a,b,c}^{=d}\right\|_{4t}^{4t} \right).$$

Proof. We rewrite $\mathbf{F}(\mathbf{x})$ into a formal sum of its homogeneous parts $\mathbf{F}(\mathbf{x}) = \sum_{d=1}^{D} \mathbf{F}^{=d}(\mathbf{x})$. By the trace inequality (see [33] Theorem 3.1), we have

$$\mathbb{E}\,\|\mathbf{F}(\mathbf{x}) - \mathbb{E}\mathbf{F}(\mathbf{x})\|_{4t}^{4t} \le D^{4t} \sum_{d=1}^{D} \mathbb{E}\left\|\mathbf{F}^{=d}(\mathbf{x}) - \mathbb{E}\mathbf{F}^{=d}(\mathbf{x})\right\|_{4t}^{4t}.$$

By Theorem 10,

$$\mathbb{E}\,\|\mathbf{F}(\mathbf{x}) - \mathbb{E}\mathbf{F}(\mathbf{x})\|_{4t}^{4t} \le D^{4t} \sum_{d=1}^{D} (48dt)^{4dt} \left(\sum_{\substack{a,b,c:\\ a+b+c=d}} L^{4ct} \left\|\mathbf{F}_{a,b,c}^{=d}\right\|_{4t}^{4t} \right). \qquad \blacktriangleleft$$

5 Gaussian Polynomial Random Matrices

In this section, we prove a moment bound for polynomial random matrices with Gaussian random variables. While the decoupling technique could work for Gaussian polynomial random matrices as well, we base our recursion framework on the following bound for its simplicity.

▶ **Lemma 12** (Polynomial Moments, see [14] Theorem 7.1). *Let* $\mathbf{H}(\mathbf{x}) : \Omega \to \mathbb{H}^d$ *be a function with* $\mathbf{x} \sim \mathcal{N}(0, \mathbb{I}_n)$. *For* $t = 1$ *and* $t \ge 1.5$,

$$\mathbb{E}\,\|\mathbf{H}(\mathbf{x}) - \mathbb{E}\mathbf{H}(\mathbf{x})\|_{2t}^{2t} \;\le\; (\sqrt{2}t)^{2t} \cdot \mathbb{E}\,\mathrm{tr}\left(\sum_{i=1}^{n} (\partial_i \mathbf{H}(\mathbf{x}))^2 \right)^{t}. \tag{3}$$

The non-Hermitian version of Lemma 12 can be easily obtained by the technique of Hermitian dilation.

▶ **Lemma 13** (Non-Hermitian Polynomial Moments). *Let* $\mathbf{F}(\mathbf{x}) : \Omega \to \mathbb{R}^{d_1 \times d_2}$ *be a function with* $\mathbf{x} \sim \mathcal{N}(0, \mathbb{I}_n)$. *For* $t = 1$ *and* $t \ge 1.5$,

$$\mathbb{E}\,\|\mathbf{F}(\mathbf{x}) - \mathbb{E}\mathbf{F}(\mathbf{x})\|_{2t}^{2t} \;\le\; (\sqrt{2}t)^{2t} \cdot \left(\mathbb{E}\left\| \left(\sum_{i=1}^{n} \partial_i \mathbf{F}(\mathbf{x})\; \partial_i \mathbf{F}^{\mathsf{T}}(\mathbf{x}) \right)^{1/2} \right\|_{2t}^{2t} + \mathbb{E}\left\| \left(\sum_{i=1}^{n} \partial_i \mathbf{F}^{\mathsf{T}}(\mathbf{x})\; \partial_i \mathbf{F}(\mathbf{x}) \right)^{1/2} \right\|_{2t}^{2t} \right).$$

▶ **Theorem 14** (Gaussian Recursion). *Let* $\mathbf{x} \sim \mathcal{N}(0, \mathbb{I}_n)$ *and* $\{\mathbf{A}_\mathbf{i}\}_{\mathbf{i} \in [n]^d}$ *be multi-indexed sequences of deterministic matrices of the same dimension. Define a degree-d homogeneous Gaussian polynomial random matrix as*

$$\mathbf{P}(\mathbf{x}) = \sum_{\mathbf{i} \in [n]^d} \left(\mathbf{A}_\mathbf{i} \cdot \prod_{j \in \{i_1, \ldots, i_d\}} x_j \right).$$

Let $a, b \in \mathbb{Z}_{\geq 0}$. Then for $2 \leq t \leq \infty$,

$$\mathbb{E}\,\|\mathbf{P}(\mathbf{x}) - \mathbb{E}\mathbf{P}(\mathbf{x})\|_{2t}^{2t} \;\leq\; (2^d\sqrt{2}t)^{2t}\left(\sum_{1\leq a+b\leq d}\|\mathbb{E}\mathbf{P}_{a,b}\|_{2t}^{2t}\right).$$

Proof. Start by applying Claim 15 with $k = 0$. Apply Claim 15 recursively until $k = d-1$ to get the desired bound. ◀

▷ Claim 15. Let $\mathbf{x} \sim \mathcal{N}(0, \mathbb{I}_n)$ and $\{\mathbf{A}_\mathbf{i}\}_{\mathbf{i}\in[n]^d}$ be multi-indexed sequences of deterministic matrices of the same dimension. Define a degree-d homogeneous Gaussian polynomial random matrix as

$$\mathbf{P}(\mathbf{x}) = \sum_{\mathbf{i}\in[n]^d}\left(\mathbf{A}_\mathbf{i}\cdot\prod_{j\in\{i_1,\ldots,i_d\}}x_j\right).$$

Let $a, b \in \mathbb{Z}_{\geq 0}$ and $a + b = k < d$. Let $\mathbf{P}_{a,b}(\mathbf{x})$ be the k-th partial derivative block matrix associated to $\mathbf{P}(\mathbf{x})$. Then for $2 \leq t \leq \infty$,

$$\begin{aligned}\mathbb{E}\,\|\mathbf{P}_{a,b} - \mathbb{E}\mathbf{P}_{a,b}\|_{2t}^{2t} \;\leq\; &(2\sqrt{2}t)^{2t}\cdot\left(\mathbb{E}\,\|\mathbf{P}_{a+1,b} - \mathbb{E}\mathbf{P}_{a+1,b}\|_{2t}^{2t} + \|\mathbb{E}\mathbf{P}_{a+1,b}\|_{2t}^{2t}\right)\\ &+ (2\sqrt{2}t)^{2t}\cdot\left(\mathbb{E}\,\|\mathbf{P}_{a,b+1} - \mathbb{E}\mathbf{P}_{a,b+1}\|_{2t}^{2t} + \|\mathbb{E}\mathbf{P}_{a,b+1}\|_{2t}^{2t}\right).\end{aligned}$$

Proof. By the non-hermitian polynomial moment (Lemma 13), we have

$$\begin{aligned}E := \mathbb{E}\,\|\mathbf{P}_{a,b} - \mathbb{E}\mathbf{P}_{a,b}\|_{2t}^{2t} &\leq (\sqrt{2}t)^{2t}\cdot\left(\mathbb{E}\left\|\left(\sum_{i=1}^n\partial_i\mathbf{P}_{a,b}\;\partial_i\mathbf{P}_{a,b}^\mathsf{T}\right)^{1/2}\right\|_{2t}^{2t} + \mathbb{E}\left\|\left(\sum_{i=1}^n\partial_i\mathbf{P}_{a,b}^\mathsf{T}\;\partial_i\mathbf{P}_{a,b}\right)^{1/2}\right\|_{2t}^{2t}\right)\\ &= (\sqrt{2}t)^{2t}\cdot\left(\mathbb{E}\,\left\|\begin{bmatrix}\partial_1\mathbf{P}_{a,b} & \ldots & \partial_n\mathbf{P}_{a,b}\end{bmatrix}\right\|_{2t}^{2t} + \mathbb{E}\left\|\begin{bmatrix}\partial_1\mathbf{P}_{a,b}\\ \vdots\\ \partial_n\mathbf{P}_{a,b}\end{bmatrix}\right\|_{2t}^{2t}\right)\\ &= (\sqrt{2}t)^{2t}\cdot\left(\mathbb{E}\,\|\mathbf{P}_{a+1,b}\|_{2t}^{2t} + \mathbb{E}\,\|\mathbf{P}_{a,b+1}\|_{2t}^{2t}\right),\end{aligned}$$

where the last equality uses our notation for partial derivative block matrices introduced in Section 2. Notice that $\mathbf{P}_{a+1,b}$ and $\mathbf{P}_{a,b+1}$ are homogeneous Gaussian polynomial random matrix of degree $k-1$. Since $\mathbf{P}_{a+1,b}$ and $\mathbf{P}_{a,b+1}$ are not necessarily centered,

$$\begin{aligned}E \;&\leq\; (\sqrt{2}t)^{2t}\cdot\left(\mathbb{E}\,\|\mathbf{P}_{a+1,b} - \mathbb{E}\mathbf{P}_{a+1,b} + \mathbb{E}\mathbf{P}_{a+1,b}\|_{2t}^{2t} + \mathbb{E}\,\|\mathbf{P}_{a,b+1} - \mathbb{E}\mathbf{P}_{a,b+1} + \mathbb{E}\mathbf{P}_{a,b+1}\|_{2t}^{2t}\right)\\ &\leq\; (2\sqrt{2}t)^{2t}\cdot\left(\mathbb{E}\,\|\mathbf{P}_{a+1,b} - \mathbb{E}\mathbf{P}_{a+1,b}\|_{2t}^{2t} + \|\mathbb{E}\mathbf{P}_{a+1,b}\|_{2t}^{2t} + \mathbb{E}\,\|\mathbf{P}_{a,b+1} - \mathbb{E}\mathbf{P}_{a,b+1}\|_{2t}^{2t} + \|\mathbb{E}\mathbf{P}_{a,b+1}\|_{2t}^{2t}\right),\end{aligned}$$

where the last inequality is due to the trace inequality (see [33] Theorem 3.1). ◁

6 Application: Graph Matrices

Denote the Erdős-Rényi random graph on n vertices as $\mathcal{G}_{n,p}$, where each edge is present with probability p independent of all other edges. When $\mathcal{G}_{n,p}$ is viewed as a probability space, it is equal to $(\Omega, \mathcal{F}, \mu)$ where Ω is the sample space of all possible graphs on n vertices. Any $G \in \Omega$ is a random vector representing all edges. Each coordinate in G, denoted by G_{ij}, is an independent random variable representing a single edge. We can construct a

random graph by drawing a copy of $G \sim \mu$. To adopt the convention in p-biased Fourier analysis, we normalize G_{ij} such that $\mathbb{E}G_{ij} = 0$ and $\mathbb{E}G_{ij}^2 = 1$, which leads to the sample space $\Omega = \left\{-\sqrt{\frac{1-p}{p}}, \sqrt{\frac{p}{1-p}}\right\}^{\binom{n}{2}}$.

▶ **Definition 16** (Shape, see e.g. [29] Definition 4.2)**.** *A shape is a tuple $\tau = (V(\tau), E(\tau), U_\tau, V_\tau)$ where $(V(\tau), E(\tau))$ is a graph and U_τ, V_τ are ordered subsets of the vertices.*

▶ **Definition 17** (Graph matrix, see e.g. [29] Definition 4.4)**.** *Given a shape τ, the associated graph matrix $\mathbf{M} : \Omega \to \mathbb{R}^{d_1 \times d_2}$ is a matrix-valued function such that*

$$\mathbf{M}[I, J] = \sum_{\substack{\varphi \in \mathcal{T}_n^k: \\ \varphi(U_\tau)=I, \varphi(V_\tau)=J}} \prod_{(u,v)\in E(\tau)} G_{\varphi(u),\varphi(v)}.$$

In other words, $\mathbf{M}$ maps an input graph $G \in \Omega$ to a $\frac{n!}{(n-|I|)!} \times \frac{n!}{(n-|J|)!}$ matrix whose rows and columns are indexed by I and J respectively.

▶ **Definition 18** (Vertex Separator, see e.g. [29] Definition 4.8)**.** *For any shape τ, a vertex separator is a subset of vertices $S \subseteq V(\tau)$ such that there is no path from U_τ to V_τ in $\tau \setminus S$, which is the shape obtained by deleting S and all edges adjacent to S. We write S_τ for a vertex separator of the minimum size.*

▷ Claim 19. For any shape τ, let $\{E_1, E_2\}$ be an arbitrary cover of $E(\tau)$. Let $S = E_1 \cap E_2$ be the vertex set of the overlapping edges between E_1 and E_2. If S contains $E_1 \cap V_\tau$ and $E_2 \cap U_\tau$, then S is a vertex separator.

Proof. Since $S = E_1 \cap E_2$, there is no path from $E_1 \backslash S$ to $E_2 \backslash S$. Additionally, if S contains $E_1 \cap V_\tau$, then $E_1 \backslash S$ is not adjacent to V_τ and $E_1 \backslash S$ contains $U_\tau \backslash S$. If S also contains $E_2 \cap U_\tau$, then $E_2 \backslash S$ is not adjacent to U_τ and $E_2 \backslash S$ contains $V_\tau \backslash S$. Since there is no path from $U_\tau \backslash S$ to $V_\tau \backslash S$, S is a vertex separator. ◁

▶ **Theorem 20.** *For any shape τ and $2 \le t \le \infty$,*

$$\mathbb{E}\,\|\mathbf{M}\|_{4t}^{4t} \;\le\; \left((48t|V(\tau)|)^{4t|V(\tau)|}(C|E(\tau)|)^{|E(\tau)|}n^{|V(\tau)|}\right)\left(\sqrt{\frac{1-p}{p}}\right)^{4t|E(S_\tau)|} n^{2t(|V(\tau)|-|S_\tau|)}$$

where C is an absolute constant and $E(S_\tau)$ are all edges adjacent to S_τ.

Proof. First, let $|V(\tau)| = k$ and we write $\mathbf{M}$ as a polynomial whose coefficients are matrices,

$$\mathbf{M} = \sum_{\varphi \in \mathcal{T}_n^k} \mathbf{B}_{\varphi(E)} \prod_{(i,j)\in E(\tau)} G_{\varphi(i),\varphi(j)},$$

where the $[I, J]$-entry of $\mathbf{B}_{\varphi(E)}$ is

$$\mathbf{B}_{\varphi(E)}[I, J] = \begin{cases} 1 & \text{if } \varphi(U_\tau) = I \text{ and } \varphi(V_\tau) = J \\ 0 & \text{otherwise.} \end{cases}$$

Note that $\mathbf{B}_{\varphi(E)}$'s have the same dimension as $\mathbf{M}$ and the rows and columns of $\mathbf{B}_{\varphi(E)}$'s are indexed in the same way as $\mathbf{M}$. Since U_τ and V_τ are ordered and each φ assigns one set of value to vertices in U_τ and V_τ, there is only one nonzero entry in each $\mathbf{B}_{\varphi(E)}$. But there might be multiple φ's for which $\mathbf{B}_{\varphi(E)}$'s are identical due to free vertices.

Let $|E(\tau)| = d$ and we rewrite $\mathbf{M}$ in a permutation symmetric way,

$$\begin{aligned}\mathbf{M} &= \sum_{\varphi\in\mathcal{T}_n^k}\left(\frac{1}{d!}\sum_{\sigma\in\mathfrak{S}_d}\mathbf{B}_{\sigma(\varphi(E))}G_{\varphi(i_{\sigma(1)}),\varphi(j_{\sigma(1)})}\ \cdots\ G_{\varphi(i_{\sigma(d)}),\varphi(j_{\sigma(d)})}\right)\\ &= \frac{1}{d!}\sum_{\varphi\in\mathcal{T}_n^k}\mathbf{A}_{\varphi(E)}\cdot G_{\varphi(i_1),\varphi(j_1)}\ \cdots\ G_{\varphi(i_d),\varphi(j_d)},\end{aligned}$$

where $\mathbf{A}_{\varphi(E)} = \sum_{\sigma\in\mathfrak{S}_d}\mathbf{B}_{\sigma(\varphi(E))}$. Notice that the indices of the random variables of $\mathbf{M}$ have the same structure as in Lemma 6, so we can decouple $\mathbf{M}$ using Lemma 6,

$$F := \mathbb{E}\,\|\mathbf{M}\|_{4t}^{4t} \ \le\ |V(\tau)|^{4t|V(\tau)|}\left(\frac{1}{d!}\right)^{4t}\cdot\mathbb{E}\left\|\sum_{\varphi\in\mathcal{T}_n^k}\mathbf{A}_{\varphi(E)}\cdot G^{(1)}_{\varphi(i_1),\varphi(j_1)}\cdots G^{(d)}_{\varphi(i_d),\varphi(j_d)}\right\|_{4t}^{4t},$$

where $\{G^{(1)},\dots,G^{(d)}\}$ denote d independent copies of G. Since $\mathbf{M}$ is a multilinear polynomial random matrix, $\mathbb{E}\mathbf{M} = \mathbf{0}$. An application of Theorem 10 yields,

$$F\ \le\ \sum_{\substack{a,b,c:\\ a+b+c=|E(\tau)|}}(48t|V(\tau)|)^{4t|V(\tau)|}\left(\frac{1}{|E(\tau)|!}\right)^{4t}\left(\sqrt{\frac{1-p}{p}}\right)^{4ct}\|\mathbf{M}_{a,b,c}\|_{4t}^{4t},$$

where $\{\mathbf{M}_{a,b,c}\}$ are partial derivative block matrices associated to $\mathbf{M}$.

Fix a set of a,b,c, $\mathbf{M}_{a,b,c}$ is a block matrix whose blocks are comprised of $\{\mathbf{A}_{\varphi(E)}\}_{\varphi\in\mathcal{T}_n^k}$. More specifically, the $[I,J]$-th block of $\mathbf{M}_{a,b,c}$ is

$$\mathbf{M}_{a,b,c}[I,J] = \partial_{G_{I_1}}\cdots\partial_{G_{I_{a+c}}}\partial_{G_{J_1}}\cdots\partial_{G_{J_b}}\mathbf{M} = \mathbf{A}_{I\cup J} = \sum_{\sigma\in\mathfrak{S}_d}\mathbf{B}_{\sigma(I\cup J)}. \tag{4}$$

Algebraically, the last equality is due to the commutativity of partial derivative operators. Combinatorially, if we fix an arbitrary ordering on the edge set $E(\tau)$, the last expression is summing up all permutations on d edges in $E(\tau)$. Now let's delve into the combinatorics. Let $E(\tau) = E_1\cup E_2$, where E_1 contains $a+c$ edges and E_2 contains $b+c$ edges with c number of overlapping edges. Let $S = E_1\cap E_2$ be the set of vertices shared by E_1 and E_2. The row and column blocks of $\mathbf{M}_{a,b,c}$ are indexed by $\varphi(E_1)$ and $\varphi(E_2)$ respectively. Let's write

$$\mathbf{M}_{a,b,c} = \sum_{\sigma\in\mathfrak{S}_d}\mathbf{M}_{a,b,c,\sigma},$$

where $\mathbf{M}_{a,b,c,\sigma}[I,J] = \mathbf{B}_{\sigma(I\cup J)}$. Applying the trace inequality (see [33] Theorem 3.1), it follows that

$$F\ \le\ \sum_{\substack{a,b,c:\\ a+b+c=|E(\tau)|}}\sum_{\sigma\in\mathfrak{S}_d}(48t|V(\tau)|)^{4t|V(\tau)|}\left(\sqrt{\frac{1-p}{p}}\right)^{4ct}\|\mathbf{M}_{a,b,c,\sigma}\|_{4t}^{4t}. \tag{5}$$

For any two distinct σ_1 and $\sigma_2\in\mathfrak{S}_d$, $\|\mathbf{M}_{a,b,c,\sigma_1}\|_{4t}^{4t} = \|\mathbf{M}_{a,b,c,\sigma_2}\|_{4t}^{4t}$ since $\mathbf{M}_{a,b,c,\sigma_1}$ and $\mathbf{M}_{a,b,c,\sigma_2}$ are identical after permuting rows and columns. So we will focus on bounding $\|\mathbf{M}_{a,b,c,\sigma_0}\|_{4t}^{4t}$ where σ_0 is the identity permutation. For identity permutation, we have $\mathbf{M}_{a,b,c,\sigma_0}[I,J] = \mathbf{B}_{I\cup J}$ by (4). Denote $\mathbf{M}_{a,b,c,\sigma_0}$ by $\mathbf{F}$,

$$\begin{aligned}E &:= \|\mathbf{M}_{a,b,c,\sigma_0}\|_{4t}^{4t} = \|\mathbf{F}\|_{4t}^{4t} = \operatorname{tr}\left(\mathbf{F}^{\mathsf{T}}\mathbf{F}\right)^{2t}\\ &= \operatorname{tr}\sum_{I_1,\dots,I_{2t}\in[n]^{2(a+c)}}\ \sum_{J_1,\dots,J_{2t}\in[n]^{2(b+c)}}\mathbf{F}^{\mathsf{T}}[I_1,J_1]\,\mathbf{F}[I_1,J_2]\,\mathbf{F}^{\mathsf{T}}[I_2,J_2]\,\mathbf{F}[I_2,J_3]\cdots\mathbf{F}^{\mathsf{T}}[I_{2t},J_{2t}]\,\mathbf{F}[I_{2t},J_1]\\ &= \operatorname{tr}\sum_{I_1,\dots,I_{2t}\in[n]^{2(a+c)}}\ \sum_{J_1,\dots,J_{2t}\in[n]^{2(b+c)}}\mathbf{B}^{\mathsf{T}}_{I_1\cup J_1}\,\mathbf{B}_{I_1\cup J_2}\,\mathbf{B}^{\mathsf{T}}_{I_2\cup J_2}\,\mathbf{B}_{I_2\cup J_3}\cdots\mathbf{B}^{\mathsf{T}}_{I_{2t}\cup J_{2t}}\,\mathbf{B}_{I_{2t}\cup J_1}\end{aligned} \tag{6}$$

First of all, $\mathbf{B}_{I_i\cup J_i} \neq \mathbf{0}$ implies that $I_i \cup J_i = \varphi(E_1 \cup E_2)$, for some $\varphi \in \mathcal{T}_n^k$ and $I_i(S) = J_i(S)$ for $1 \le i \le 2t$. Similarly, $\mathbf{B}_{I_i\cup J_{i+1}} \neq \mathbf{0}$ implies that $I_i \cup J_{i+1} = \varphi'(E_1 \cup E_2)$, for some $\varphi' \in \mathcal{T}_n^k$ and $I_i(S) = J_{i+1}(S)$ for $1 \le i \le 2t$ (the additions in the subscripts are in mod $2t$, *i.e.*, $J_{2t+1} = J_1$). Secondly, for each $\mathbf{B}_{I_i\cup J_i} \neq \mathbf{0}$, there is only one nonzero entry, namely $\mathbf{B}_{I_i\cup J_i}[I_i \cup J_i(U_\tau), I_i \cup J_i(V_\tau)] = 1$. So $\mathbf{B}_{I_i\cup J_i}^\mathsf{T}\mathbf{B}_{I_i\cup J_{i+1}} \neq \mathbf{0}$ if $I_i \cup J_i(U_\tau) = I_i \cup J_{i+1}(U_\tau)$ and the only nonzero entry is

$$\mathbf{B}_{I_i\cup J_i}{}^\mathsf{T}\mathbf{B}_{I_i\cup J_{i+1}}[I_i \cup J_i(V_\tau), I_i \cup J_{i+1}(V_\tau)] = 1.$$

It follows that $\mathbf{B}_{I_i\cup J_i}^\mathsf{T}\mathbf{B}_{I_i\cup J_{i+1}} \neq \mathbf{0}$ for $1 \le i \le 2t$ if $J_1(E_2\cap U_\tau) = J_2(E_2\cap U_\tau) = \cdots = J_{2t}(E_2\cap U_\tau)$. Lastly, $\mathbf{B}_{I_i\cup J_i}^\mathsf{T}\mathbf{B}_{I_i\cup J_{i+1}}\mathbf{B}_{I_{i+1}\cup J_{i+1}}^\mathsf{T}\mathbf{B}_{I_{i+1}\cup J_{i+2}} \neq \mathbf{0}$ if $I_i \cup J_{i+1}(V_\tau) = I_{i+1} \cup J_{i+1}(V_\tau)$ and the only nonzero entry is

$$\mathbf{B}_{I_i\cup J_i}^\mathsf{T}\mathbf{B}_{I_i\cup J_{i+1}}\mathbf{B}_{I_{i+1}\cup J_{i+1}}^\mathsf{T}\mathbf{B}_{I_{i+1}\cup J_{i+2}}[I_i \cup J_i(V_\tau), I_{i+1} \cup J_{i+2}(V_\tau)] = 1.$$

It follows that $\mathbf{B}_{I_1\cup J_1}^\mathsf{T}\ \mathbf{B}_{I_1\cup J_2}\ \mathbf{B}_{I_2\cup J_2}^\mathsf{T}\ \mathbf{B}_{I_2\cup J_3}\cdots\mathbf{B}_{I_{2t}\cup J_{2t}}^\mathsf{T}\ \mathbf{B}_{I_{2t}\cup J_1} \neq \mathbf{0}$ if $I_1(E_1 \cap V_\tau) = I_2(E_1 \cap V_\tau) = \cdots = I_{2t}(E_1 \cap V_\tau)$ and the only nonzero entry is

$$\mathbf{B}_{I_1\cup J_1}^\mathsf{T}\ \mathbf{B}_{I_1\cup J_2}\ \mathbf{B}_{I_2\cup J_2}^\mathsf{T}\ \mathbf{B}_{I_2\cup J_3}\cdots\mathbf{B}_{I_{2t}\cup J_{2t}}^\mathsf{T}\ \mathbf{B}_{I_{2t}\cup J_1}[I_1 \cup J_1(V_\tau), I_{2t} \cup J_1(V_\tau)] = 1.$$

Since $I_1(E_1 \cap V_\tau) = I_{2t}(E_1 \cap V_\tau)$, we have $I_1 \cup J_1(V_\tau) = I_{2t} \cup J_1(V_\tau)$. So if the summand is nonzero, it has a 1 on its diagonal.

To summarize, each summand in (6) is nonzero if and only if $I_i \cup J_i = \varphi_i(E_1 \cup E_2), I_i \cup J_{i+1} = \varphi_i'(E_1 \cup E_2)$ for some $\varphi_i, \varphi_i' \in \mathcal{T}_n^k$, I_i's and J_i's agree on S, I_i's agree on $E_1 \cap V_\tau$ and J_i's agree on $E_2 \cap U_\tau$ for $1 \le i \le 2t$. Since each nonzero summand contributes a 1 on the diagonal, we can simply count the number of nonzero summands to compute the trace in (6). Notice that the number of nonzero summands is equal to the number of $\varphi_1, \ldots, \varphi_{2t}, \varphi_1', \ldots, \varphi_{2t}'$ that satisfy all the constraints. Hence

$$E \le n^{|S|}\ n^{|E_1\cap V_\tau|}\ n^{|E_2\cap U_\tau|}\ n^{2t|E_1\setminus V_\tau\setminus S|}\ n^{2t|E_2\setminus U_\tau\setminus S|}. \tag{7}$$

For a fixed set of a, b, c, (7) provides an upper bound for $\|\mathbf{M}_{a,b,c,\sigma}\|_{4t}^{4t}$ for any permutation $\sigma \in \mathfrak{S}_d$. But what is an upper bound for $\|\mathbf{M}_{a,b,c,\sigma}\|_{4t}^{4t}$ among all possible sets of a, b, c? By varying a, b, c, we can always make $S = E_1 \cap E_2$ large enough to contain vertices in $E_1 \cap V_\tau$ and $E_2 \cap U_\tau$. Thus S is a vertex separator by Claim 19. It follows that $E_1\setminus V_\tau\setminus S = E_1\setminus S$ and $E_2\setminus U_\tau\setminus S = E_2\setminus S$. Hence

$$E \ \le\ n^{|V(\tau)|}\ n^{2t|E_1\setminus S|}\ n^{2t|E_2\setminus S|} \ \le\ n^{|V(\tau)|}\ n^{2t(|V(\tau)|-|S_\tau|)}, \tag{8}$$

where S_τ is a vertex separator of minimum size. Substituting (8) into (5) yields the desired bound. ◀

▶ **Corollary 21.** *For any given shape τ, any $\varepsilon > 0$, with probability $1 - \varepsilon$,*

$$\|\mathbf{M}\| \ \le\ |V(\tau)|^{|V(\tau)|}\left(C\log\left(|E(\tau)|^{|E(\tau)|}n^{|V(\tau)|}/\varepsilon\right)\right)^{|V(\tau)|}\left(\sqrt{\frac{1-p}{p}}\right)^{|E(S_\tau)|}\sqrt{n}^{|V(\tau)|-|S_\tau|},$$

where $C > 0$ is an absolute constant.

Proof. By Markov's inequality and Theorem 20, we have

$$\begin{aligned}\mathbb{P}(\|\mathbf{M}\| \ge \theta) \ &\le\ \mathbb{P}(\|\mathbf{M}\|_{4t}^{4t} \ge \theta^{4t}) \ \le\ \theta^{-4t}\mathbb{E}\,\|\mathbf{M}\|_{4t}^{4t} \\ &= \theta^{-4t}\left((48t|V(\tau)|)^{4t|V(\tau)|}(C|E(\tau)|)^{|E(\tau)|}n^{|V(\tau)|}\right)\left(\sqrt{\frac{1-p}{p}}\right)^{4t|E(S_\tau)|}n^{2t(|V(\tau)|-|S_\tau|)}. \end{aligned}\tag{9}$$

Set the right hand side of (9) to ε by taking

$$\theta = \left(\varepsilon^{-\frac{1}{4t}}(48t|V(\tau)|)^{|V(\tau)|}(C|E(\tau)|)^{\frac{|E(\tau)|}{4t}} n^{\frac{|V(\tau)|}{4t}}\right)\left(\sqrt{\frac{1-p}{p}}\right)^{|E(S_\tau)|}\sqrt{n}^{|V(\tau)|-|S_\tau|}.$$

Take

$$t = \frac{1}{4}\log\left(|E(\tau)|^{|E(\tau)|}n^{|V(\tau)|}/\varepsilon\right)$$

to complete the proof. ◀

References

1. Radosław Adamczak and Paweł Wolff. Concentration inequalities for non-lipschitz functions with bounded derivatives of higher order. *Probability Theory and Related Fields*, 162(3):531–586, 2015.
2. Kwangjun Ahn, Dhruv Medarametla, and Aaron Potechin. Graph matrices: Norm bounds and applications. *arXiv preprint*, 2021. `arXiv:1604.03423`.
3. Richard Aoun, Marwa Banna, and Pierre Youssef. Matrix poincaré inequalities and concentration. *Advances in Mathematics*, 371:107251, 2020.
4. Mitali Bafna, Jun-Ting Hsieh, Pravesh K. Kothari, and Jeff Xu. Polynomial-time power-sum decomposition of polynomials. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 956–967, 2022. `doi:10.1109/FOCS54457.2022.00094`.
5. Afonso S Bandeira, March T Boedihardjo, and Ramon van Handel. Matrix concentration inequalities and free probability. *Inventiones Mathematicae*, pages 1–69, 2023.
6. Nikhil Bansal, Haotian Jiang, and Raghu Meka. Resolving matrix Spencer conjecture up to poly-logarithmic rank. In *Proceedings of the 55th ACM Symposium on Theory of Computing*, pages 1814–1819, 2023. `doi:10.1145/3564246.3585103`.
7. Boaz Barak, Samuel Hopkins, Jonathan Kelner, Pravesh K Kothari, Ankur Moitra, and Aaron Potechin. A nearly tight sum-of-squares lower bound for the planted clique problem. *SIAM Journal on Computing*, 48(2):687–735, 2019. `doi:10.1137/17M1138236`.
8. Sergey G Bobkov, Friedrich Götze, and Holger Sambale. Higher order concentration of measure. *Communications in Contemporary Mathematics*, 21(03):1850043, 2019.
9. Tatiana Brailovskaya and Ramon van Handel. Universality and sharp matrix concentration inequalities. *arXiv preprint*, 2022. `arXiv:2201.05142`.
10. Jingqiu Ding, Tommaso d'Orsi, Chih-Hung Liu, David Steurer, and Stefan Tiegel. Fast algorithm for overcomplete order-3 tensor decomposition. In *Conference on Learning Theory*, pages 3741–3799. PMLR, 2022. URL: `https://proceedings.mlr.press/v178/ding22a.html`.
11. Rong Ge, Qingqing Huang, and Sham M. Kakade. Learning mixtures of gaussians in high dimensions. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 761–770, New York, NY, USA, 2015. Association for Computing Machinery. `doi:10.1145/2746539.2746616`.
12. Samuel B Hopkins, Tselil Schramm, and Jonathan Shi. A robust spectral algorithm for overcomplete tensor decomposition. In *Conference on Learning Theory*, pages 1683–1722. PMLR, 2019. URL: `http://proceedings.mlr.press/v99/hopkins19b.html`.
13. Samuel B Hopkins, Tselil Schramm, Jonathan Shi, and David Steurer. Fast spectral algorithms from sum-of-squares proofs: tensor decomposition and planted sparse vectors. In *Proceedings of the 48th ACM Symposium on Theory of Computing*, pages 178–191, 2016. `doi:10.1145/2897518.2897529`.
14. De Huang and Joel A. Tropp. From poincaré inequalities to nonlinear matrix concentration. *Bernoulli*, 2020.
15. De Huang and Joel A. Tropp. Nonlinear matrix concentration via semigroup methods. *Electronic Journal of Probability*, 26:Art. No. 8, January 2021.

16 Chris Jones, Aaron Potechin, Goutham Rajendran, Madhur Tulsiani, and Jeff Xu. Sum-of-squares lower bounds for sparse independent set. In *Proceedings of the 62nd IEEE Symposium on Foundations of Computer Science*, 2021.

17 Jeong Han Kim and Van H Vu. Concentration of multivariate polynomials and its applications. *Combinatorica*, 20(3):417–434, 2000. doi:10.1007/S004930070014.

18 Bohdan Kivva and Aaron Potechin. Exact nuclear norm, completion and decomposition for random overcomplete tensors via degree-4 sos. *arXiv preprint arXiv:2011.09416*, 2020. arXiv:2011.09416.

19 Stanislaw Kwapien. Decoupling Inequalities for Polynomial Chaos. *The Annals of Probability*, 15(3):1062–1071, 1987.

20 Cécilia Lancien and Pierre Youssef. A note on quantum expanders, 2023. arXiv:2302.07772.

21 Rafał Latała. Estimates of moments and tails of gaussian chaoses. *The Annals of Probability*, 34(6):2315–2331, 2006.

22 Lester Mackey, Michael Jordan, Richard Chen, Brendan Farrell, and Joel Tropp. Matrix concentration inequalities via the method of exchangeable pairs. *The Annals of Probability*, 42, 2012.

23 Terry R. McConnell and Murad S. Taqqu. Double integration with respect to symmetric stable processes. Technical report, Technical Report 618, Cornell Univ., 1984.

24 Ankur Moitra and Alexander S Wein. Spectral methods from tensor networks. In *Proceedings of the 51st ACM Symposium on Theory of Computing*, pages 926–937, 2019. doi:10.1145/3313276.3316357.

25 Ryan O'Donnell and Yu Zhao. Polynomial bounds for decoupling, with applications. In *Proceedings of the 31st Conference on Computational Complexity*, CCC '16, Dagstuhl, DEU, 2016. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2016.24.

26 Daniel Paulin, Lester Mackey, and Joel A. Tropp. Efron–Stein inequalities for random matrices. *The Annals of Probability*, 44(5):3431–3473, 2016.

27 Víctor H. Peña and Evarist Giné. *Decoupling: From dependence to independence*. Springer-Verlag, 1999.

28 Goutham Rajendran. *Nonlinear Random Matrices and Applications to the Sum of Squares Hierarchy*. PhD thesis, University of Chicago, 2022.

29 Goutham Rajendran and Madhur Tulsiani. Concentration of polynomial random matrices via efron-stein inequalities. *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2023.

30 Holger Rauhut. Compressive sensing and structured random matrices. In *Theoretical Foundations and Numerical Methods for Sparse Recovery*, pages 1–92. De Gruyter, Berlin, New York, 2010.

31 Warren Schudy and Maxim Sviridenko. Bernstein-like concentration and moment inequalities for polynomials of independent random variables: multilinear case. *arXiv preprint*, 2011. arXiv:1109.5193.

32 Warren Schudy and Maxim Sviridenko. Concentration and moment inequalities for polynomials of independent random variables. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 437–446. SIAM, 2012. doi:10.1137/1.9781611973099.37.

33 Khalid Shebrawi and Hussien Albadawi. Trace inequalities for matrices. *Bulletin of the Australian Mathematical Society*, 87(1):139–148, 2013.

34 Joel A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8(1-2):1–230, 2015.

35 Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.

Listing 6-Cycles in Sparse Graphs

Virginia Vassilevska Williams
MIT, Cambridge, MA, USA

Alek Westover
MIT, Cambridge, MS, USA

Abstract

This work considers the problem of output-sensitive listing of occurrences of $2k$-cycles for fixed constant $k \geq 2$ in an undirected host graph with m edges and t $2k$-cycles. Recent work of Jin and Xu (and independently Abboud, Khoury, Leibowitz, and Safier) [STOC 2023] gives an $O(m^{4/3} + t)$ time algorithm for listing 4-cycles, and recent work by Jin, Vassilevska Williams and Zhou [SOSA 2024] gives an $\widetilde{O}(n^2 + t)$ time algorithm for listing 6-cycles in n node graphs. We focus on resolving the next natural question: obtaining listing algorithms for 6-cycles in the sparse setting, i.e., in terms of m rather than n. Previously, the best known result here is the better of Jin, Vassilevska Williams and Zhou's $\widetilde{O}(n^2 + t)$ algorithm and Alon, Yuster and Zwick's $O(m^{5/3} + t)$ algorithm.

We give an algorithm for listing 6-cycles with running time $\widetilde{O}(m^{1.6} + t)$. Our algorithm is a natural extension of Dahlgaard, Knudsen and Stöckel's [STOC 2017] algorithm for detecting a $2k$-cycle. Our main technical contribution is the analysis of the algorithm which involves a type of "supersaturation" lemma relating the number of $2k$-cycles in a bipartite graph to the sizes of the parts in the bipartition and the number of edges. We also give a simplified analysis of Dahlgaard, Knudsen and Stöckel's $2k$-cycle detection algorithm (with a small polylogarithmic increase in the running time), which is helpful in analyzing our listing algorithm.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis

Keywords and phrases Graph algorithms, cycles listing, fine-grained complexity, sparse graphs

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.92

Related Version *arXiv Version*: `https://arxiv.org/abs/2411.07499`

Supplementary Material *Software*: `https://github.com/awestover/listing-C6s-LP` [18] archived at `swh:1:dir:72ea68e559495cac21c615e3c3bfa8cf3a18653f`

Funding *Virginia Vassilevska Williams*: Supported by NSF Grant CCF-2330048, BSF Grant 2020356 and a Simons Investigator Award.

Acknowledgements We'd like to thank Nathan S. Sheffield, Claire Zhang, and Ce Jin for helpful discussions.

1 Introduction

Listing copies of a small pattern graph that occur as subgraphs of a host graph is a fundamental problem in algorithmic graph theory. In this work, we consider the problem of listing C_{2k}'s (i.e., $2k$-cycles) for fixed constant $k \geq 2$. Some examples of applications of C_{2k} listing include analyzing social networks [9], and understanding causal relationships in biological interaction graphs [14]. (See e.g. [19] for further motivation.) In the following discussion we consider an n vertex m edge graph G with t C_{2k}'s (where $k \geq 2$ will be clear from context and t is not necessarily known).

The main reason for focusing on even length cycles is that while there are dense graphs that contain no odd cycles (e.g. bipartite graphs), a classic result of Bondy and Simonovits [6] states that any graph with at least $100kn^{1+1/k}$ edges contains a C_{2k}, for any integer

© Virginia Vassilevska Williams and Alek Westover;
licensed under Creative Commons License CC-BY 4.0

$k \geq 2$. This fact enables efficient "combinatorial" algorithms for C_{2k} detection, in contrast to the case of odd length cycles where efficient C_{2k+1} detection algorithms are based on matrix multiplication. In fact, very simple reductions (e.g. [16]) show that for any $k \geq 1$, C_{2k+1} detection is at least as hard as triangle detection, and the latter problem is known to be fine-grained subcubically equivalent to Boolean Matrix Multiplication [17]. Thus, fast algorithms for odd cycles cannot avoid matrix multiplication. We focus on even cycles from now on.

A classic result of Yuster and Zwick [19] shows how to determine whether a graph contains a C_{2k} in $O(n^2)$ time for any given constant $k \geq 2$. This quadratic running time is conjectured to be optimal in general (see [2, 15, 8], also discussed below); in particular, [2] gave concrete evidence for the hardness of C_4 detection.

Nevertheless, in the regime of sparser graphs, where $m < o(n^{1+1/k})$ improvements are possible. For C_4's there is a simple $O(m^{4/3})$ algorithm [5] that matches the performance of the $O(n^2)$ algorithm of [19] for $m = \Theta(n^{3/2})$, and improves on the performance for all $m < o(n^{3/2})$. Dahlgaard, Knudsen and Stöckel [8] give an algorithm for C_{2k} detection with running time $O(m^{2k/(k+1)})$ for every constant $k \geq 2$. This matches the $O(n^2)$ running time from [19] for $m = \Theta(n^{1+1/k})$ and improves on it for $m < o(n^{1+1/k})$.

In fine-grained complexity, it is common to assume widely-believed hypotheses and use reductions to obtain conditional lower bounds for fundamental problems. For the special case of cycle detection problems, most conditional lower bounds are based on hypotheses related to triangle detection. One of the most common hypotheses is that triangle detection in n-node graphs does not have a "combinatorial" [1] $O(n^{3-\varepsilon})$ time algorithm for $\varepsilon > 0$ (in the word-RAM model). Vassilevska W. and Williams [17] showed that this hypothesis is equivalent to the so called BMM Hypothesis that postulates that there is no $O(n^{3-\varepsilon})$ time combinatorial algorithm for multiplying two $n \times n$ *Boolean* matrices.

As mentioned earlier, it is easy to show that under the BMM hypothesis, detecting any odd cycle C_{2k+1} requires $n^{3-o(1)}$ time. Dahlgaard, Knudsen and Stöckel [8] gave lower bounds for combinatorial detection of even cycles of fixed length in *sparse* graphs. They show that there is no combinatorial algorithm for C_6 detection, or C_{2k} detection for any $k > 4$, with running time $O(m^{3/2-\varepsilon})$ for $\varepsilon > 0$. Lincoln and Vyas [15] give a similar conditional lower bound under a different hypothesis, extending the lower bounds for potentially non-combinatorial algorithms. Lincoln and Vyas show that, for large enough constant k, a C_{2k} detection algorithm in graphs with $m \leq O(n)$ with running time $m^{3/2-\varepsilon}$ would imply an algorithm for max-3-SAT with running time $2^{(1-\varepsilon')n}n^{O(1)}$.

The problem of *listing* even cycles is less well-understood. Without improving the C_{2k} detection algorithms discussed above, the best result that we could hope for in general is an algorithm with running time $O(n^2 + t)$, where t is the number of $2k$-cycles in the graph, and $O(m^{2k/(k+1)} + t)$ in the sparse setting where $m < o(n^{1+1/k})$. Recently, [3] and [12] gave such an algorithm for C_4 listing. In fact, Jin and Xu's [12] algorithm gives a stronger guarantee: After $O(m^{4/3})$ pre-processing time, they can (deterministically) enumerate 4-cycles with $O(1)$ delay per cycle. Jin and Xu [12] (and concurrently by Abboud, Bringmann, and Fischer [1]) shows that, under the 3SUM Hypothesis, there is no algorithm for C_4 enumeration with $O(n^{2-\varepsilon})$ or $O(m^{4/3-\varepsilon})$ pre-processing time and $n^{o(1)}$ delay, for $\varepsilon > 0$.

[1] The class of combinatorial algorithms is not well defined, but intuitively refers to algorithms that do not use fast matrix multiplication as a subroutine.

For C_6's, Jin, Zhou and Vassilevska Williams [11] give an $\widetilde{O}(n^2+t)$ listing algorithm. For sparse listing algorithms, the previous state of the art is [5] whose work implies an $O(m^{(2k-1)/k}+t)$ time C_{2k} listing algorithm. See also [7] where the complexity of the harder problem of listing H-partite[2] subgraphs is investigated.

1.1 Our contributions

We consider the problem of listing all C_6s in an m-edge graph. The best known result so far is an $O(m^{5/3}+t)$ time algorithm that follows from the work of [5]. Our main result is the first improvement over this 27 year old running time:

▶ **Theorem 23.** *There is an algorithm for listing C_6's in time $\widetilde{O}(m^{1.6}+t)$.*

We now summarize the ideas needed to prove Theorem 23.

Theorem 23 follows easily after establishing a certain bound on the number of **capped k-walks** in a graph. Capped k-walks are a notion introduced by Dahlgaard, Knudsen and Stöckel in [8]. Roughly speaking, a capped k-walk is a walk of length k where the first vertex in the walk has higher degree than the remaining vertices in the walk (handling vertices of equal degree requires a bit of additional care). [8]'s $O(m^{2k/(k+1)})$ time C_{2k} detection algorithm is based on the following fact:

▶ **Theorem 4** ([8]). *Let G be a C_{2k}-free graph with maximum degree $\Delta(G) \leq m^{2/(k+1)}$. Then, there are at most $\widetilde{O}(m^{2k/(k+1)})$ capped k-walks in G.*

One of our main contributions is a *simplified proof* of Theorem 4, although with polylogarithmically worse guarantees than [8]. This simplified analysis of capped k-walks is quite helpful in obtaining Theorem 23. The key lemma used to prove Theorem 4 is a generalization to bipartite graphs of Bondy and Simonovits' classic theorem on the extremal number of C_{2k}'s. Specifically, the result is:

▶ **Theorem 1** ([8]). *Let G be a bipartite graph with vertex parts of sizes L, R and with m edges. If $m > 100k(L+R+(LR)^{(k+1)/(2k)})$ then G contains a C_{2k}.*

In order to use capped k-walks for a listing algorithm, it would be useful to have a **supersaturation** variant of Theorem 1, which would guarantee the presence of *many* C_{2k}'s if the edge density is large; the fact that this supersaturation result could help with listing was communicated to us by Jin and Zhou [13]. The supersaturation analog of Bondy and Simonovits' [6] extremal number for C_{2k}'s is known:

▶ **Theorem 2** ([10]). *For every integer $k \geq 2$, there exist constants c, C such that if G is an n-node graph with $m \geq Cn^{1+1/k}$ edges, then G contains at least $c(m/n)^{2k}$ copies of C_{2k}.*

Jin and Zhou [13] formulated the following conjectured generalization of Theorem 2:

▶ **Conjecture 25.** (The Unbalanced Supersaturation Conjecture [Jin and Zhou'24]) *Let G be an m edge bipartite graph with vertex bipartition $A \sqcup B$, with t C_{2k}'s. Suppose $m \geq 100k(|A|+|B|+(|A||B|)^{(k+1)/2k})$. Then,*

$$t \geq \Omega\left(\frac{m^{2k}}{|A|^k|B|^k}\right).$$

[2] A k-node H is an H-partite subgraph of a k-partite graph G with vertex set $V = \cup_{a\in V(H)} V_a$ if there are k vertices $v_1, \ldots, v_k$ such that $v_a \in V_a$ for each $a \in \{1, \ldots, k\}$ and the mapping $a \to v_a$ is an isomorphism between H and the subgraph of G induced by $v_1, \ldots, v_k$. In other words, instead of looking for an arbitrary subgraph of G isomorphic to H, one only focuses on the subgraphs with exactly one node in each V_a and such that the node picked from V_a corresponds to node a of H.

Jin and Zhou obtained the following conclusion using the approach of [8]:

▶ **Fact 3** ([13]). If the Unbalanced Supersaturation Conjecture is true, then there is an $\widetilde{O}(t + m^{2k/(k+1)})$ time algorithm for $2k$-cycle listing for all $k \geq 2$.

In Corollary 29 we give a simple proof of Jin and Zhou's fact above. Thus, an approach to obtaining the conjectured optimal running time of $\widetilde{O}(t + m^{2k/(k+1)})$ for C_{2k}-listing would be to prove Conjecture 25. Unfortunately, establishing or refuting Conjecture 25 remains a challenging open problem.

Our main technical contribution is a proof of a weaker version of Conjecture 25 for $k = 3$. Then, we show that this partial progress towards Conjecture 25 can be used in our simplified method for analyzing capped k-walks to obtain a bound on the number of capped k-walks, and consequentially an improved C_6 listing algorithm in the sparse setting (namely, Theorem 23).

1.2 Open Questions

Proving or refuting Conjecture 25 is an important open question. It also is valuable to consider other avenues towards obtaining C_{2k} listing algorithms, especially in case Conjecture 25 turns out to be false. The listing algorithms of [11], [12], [3] use a variety of combinatorial insights that could potentially be generalized to larger k. It is also possible that a hybrid approach is productive: one can first use progress towards proving Conjecture 25 to force the instance to have a specific structure, and then use different combinatorial insights to solve the structured version of the problem.

1.3 Paper Outline

In Section 2 we present our simplified analysis of [8]'s C_{2k} detection algorithm. In Section A we show how to modify our simple analysis of [8]'s C_{2k} detection algorithm to get a C_{2k} listing algorithm, assuming the Unbalanced Supersaturation Conjecture. In Section 3 we present progress towards resolving the Unbalanced Supersaturation Conjecture. In Section 4 we show how to use our progress towards Conjecture 25 in our simplified capped k-walk analysis to obtain a listing algorithm for C_6's with running time $\widetilde{O}(m^{1.6} + t)$.

1.4 Notations

We use $\Delta(G)$ to denote the maximum degree of graph G. We write $N(v)$ to denote the neighborhood of vertex v, and for $S \subseteq V(G)$ we write $N_S(v)$ to denote $N(v) \cap S$. For graph G, we will use $V(G), E(G)$ to denote the vertex and edge sets of G. When G is clear from the context we also denote $V(G)$ by V and $E(G)$ by E. We let $n = |V|, m = |E|$ (when the graph being discussed is clear), and assume $m \geq n$. For vertex subsets A, B we write $e(A, B)$ to denote $|E \cap (A \times B)|$. A walk / path of length k will refer to a walk / path with k edges. Given $A, B \subseteq V$, we will write $G[A, B]$ to denote the induced subgraph on A, B, i.e., a graph with vertex set $A \cup B$ and edge set $E(G) \cap (A \times B)$. We will write $G[A]$ to denote $G[A, A]$.

We write $[x]$ to denote the set $\{1, 2, \ldots, \lceil x \rceil\}$. We use log to denote the base-2 logarithm. Define the **dyadic intervals** $\mathcal{I}_n$ as follows: letting n' denote the power of two in $[n, 2n)$, we define $\mathcal{I}_n = (\{0\}, \{1\}, [2, 4), [4, 8), \ldots, [n'/4, n'/2), [n'/2, n'])$. Note that $|\mathcal{I}_n| = 1 + \log n'$. For $j \in [\log n']$, the j-th dyadic interval refers to the j-th set in the list $\mathcal{I}_n$ of dyadic intervals, *starting counting from* $\{1\}$, or in other words *excluding the set* $\{0\}$. That is, the j-th dyadic interval is $[2^{j-1}, 2^j)$ for $j \in [(\log n') - 1]$ and the $(\log n')$-th one is $[n'/2, n']$.

2 A Simpler Analysis of the DKS C_{2k} Detection Algorithm

Dahlgaard, Knudsen and Stöckel [8] define **capped k-walks** and give a sophisticated analysis to bound the number of capped k-walks in C_{2k}-free graphs (assuming $\Delta(G) \leq m^{2/(k+1)}$). In this section we provide a simpler – although quantitatively worse by logarithmic factors – version of their analysis. This will be useful in future sections when we analyze the number of capped k-walks in graphs that are not C_{2k}-free.

Given a total ordering $\succ$ of the vertices in a graph, a **$\succ$-capped k-walk** [8] is a walk $x_0, x_1, \ldots, x_k$ where $x_0 \succ x_i$ for all $i \in [k]$. Throughout the paper we will assume implicitly an ordering $\succ$ of the vertices that satisfies $v_i \succ v_j$ whenever $\deg(v_i) > \deg(v_j)$ (but is otherwise arbitrary). Thus, we will refer simply to capped k-walks (leaving the dependence on $\succ$ implicit). The main result of this section is a simplified proof of the following fact:

▶ **Theorem 4** ([8]). *Let G be a C_{2k}-free graph with maximum degree $\Delta(G) \leq m^{2/(k+1)}$. Then, there are at most $\widetilde{O}(m^{2k/(k+1)})$ capped k-walks in G.*

We note that if G were (m/n)-regular then it would contain $\Theta(n(m/n)^k) \leq O(m^{2k/(k+1)})$ k-walks due to $m \leq O(n^{1+1/k})$ (which holds because G is C_{2k}-free). While a non-regular graph G may have more than $O(m^{2k/(k+1)})$ many k-walks, Theorem 4 states that G does not have more than this many *capped* k-walks (assuming $\Delta(G) \leq m^{2/(k+1)}$). At a high level, the proof of the bound on capped k-walks follows from applying the bipartite generalization of Bondy and Simonovits' extremal number bound [6] (Theorem 1) to a certain **layered organization** of the graph with nice regularity properties. This layered organization of the graph is our contribution.

In [8], the authors took a different approach: they defined the "ϕ-norm" of a vector v as

$$||v||_\phi = \int_0^\infty \sqrt{|\{i \mid |v_i| \geq x\}|}dx.$$

They then related the ϕ-norm of $X_G^k \mathbb{1}$, (where X_G is the graph's adjacency matrix and $\mathbb{1}$ is the all-ones vector) to the number of capped k-walks in G, and analyzed a large set of inequalities using combinatorial insights to bound this ϕ-norm. We find that the layered organization of the graph technique is more helpful than the ϕ-norm approach in elucidating [8]'s elegant result, and that this layered organization technique is easier to use in bounding the number of capped k-walks in graphs which are not C_{2k}-free. Our layered organization of the graph is depicted in Figure 1, and formally defined and analyzed in Lemma 5.

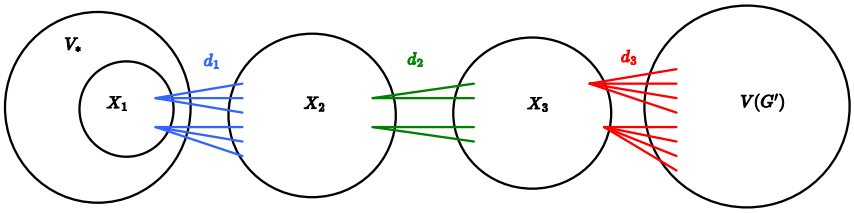


Figure 1 A layer pattern capturing many capped 3-walks.

▶ **Lemma 5.** *Fix graph G. There exists $d_* \in [n]$ such that if we let V_* denote the set of vertices with degree between $[d_*/2, d_*]$ and let G' denote the induced subgraph of G on vertices with degree at most d_*, then there exist $(X_1, \ldots, X_k) \subseteq V_* \times V(G')^{k-1}, (d_1, \ldots, d_k) \in \mathbb{N}^k$ such that letting $X_{k+1} = V(G')$,*

1. *The number of capped k-walks in G is at most $O(\log^{k+1} n)$ times the number of k-walks in $X_1 \times X_2 \times \cdots \times X_k \times V(G')$,*
2. *For each $i \in [k]$, every $v \in X_i$ has $|N_{X_{i+1}}(v)| \in [d_i/2, d_i]$.*

Proof. For $j \in [\log n]$, let V_j denote the set of vertices $v \in V$ with degree lying in the j-th dyadic interval. Fix j maximizing the number of capped k-walks in G that start in a vertex in V_j. Let $V_* = V_j, d_* = 2^j$, and let G' be the induced subgraph on vertices of degree at most d_*. The number of capped k-walks in G is at most $O(\log n)$ times the number of k-walks in G' that start in V_*; this is because the vertices in $V(G) \setminus V(G')$ are of too high degree to participate in a capped k-walk starting in V_*, and because every capped k-walk must start in some V_j.

For each $i_k \in [\log n]$ let $W^{(k)}_{i_k}$ denote the set of vertices in G' whose degree in G' lies in the i_k-th dyadic interval. Now, iteratively for $\ell = k-1, k-2, \ldots, 2$, for each $i_\ell \in [\log n]$ define $W^{(\ell)}_{i_k, i_{k-1}, \ldots, i_\ell}$ to be the set of vertices $v \in V(G')$ such that the number of neighbors of v in $W^{(\ell+1)}_{i_k, i_{k-1}, \ldots, i_{\ell+1}}$ lies in the i_ℓ-th dyadic interval. Finally, for each $i_1 \in [\log n]$ define $W^{(1)}_{i_k, i_{k-1}, \ldots, i_1}$ to be the set of vertices $v \in V_*$ such that the number of neighbors of v in $W^{(2)}_{i_k, i_{k-1}, \ldots, i_2}$ lies in the i_1-th dyadic interval.

Intuitively, looking at Figure 1, we are building up the sets X_ℓ from right to left. Here we think of $W^{(\ell)}_{i_k, i_{k-1}, \ldots, i_\ell}$ as representing X_ℓ for some choice of $i_k, i_{k-1}, \ldots, i_\ell$ that we will fix shortly.

Now, fix a choice $(i_k, \ldots, i_1) \in [\log n]^k$ that maximizes the number of k-walks

$$(x_1, \ldots, x_{k+1}) \in W^{(1)}_{i_k, \ldots, i_1} \times W^{(2)}_{i_k, \ldots, i_2} \times \ldots, W^{(k)}_{i_k} \times V(G'). \tag{1}$$

Note that every k-walk in G' starting in V_* must be of the form described in Equation (1) for *some* $(i_k, \ldots, i_1) \in [\log n]^k$. Thus, for our choice of $(i_k, \ldots, i_1)$, at least an $\Omega(1/\log^k n)$-fraction of the k-walks in G' starting in V_* are of the form Equation (1). For this choice of $(i_k, \ldots, i_1)$, we define for each $j \in [\log n]$,

$$X_j = W^{(j)}_{i_k, \ldots, i_j}, \quad d_j = 2^{i_j}.$$

These are the sets X_j, d_j that we wanted in the lemma statement, and they satisfy the required degree regularity property. Also because an $\Omega(1/\log^k n)$-fraction of the k-walks in G' starting in V_* are of the form Equation (1), and because the number of capped k-walks in G is at most $O(\log n)$ times the number of k-walks in G' starting in V_*, we have that the number of capped k-walks in G is at most $\widetilde{O}(1)$ times the number of k-walks of the form Equation (1). ◀

We have shown in Lemma 5 that in order to bound the number of capped-k walks in G, it suffices to bound the number of k-walks where each vertex of the k-walk is required to belong to a specific "layer" set, where consecutive layers satisfy a degree regularity property. This motivates studying how large an edge density we can have between two such consecutive layers while avoiding C_{2k}'s. First, we need a minor modification of Theorem 1.

▶ **Corollary 6.** *Let G be a C_{2k}-free graph. Fix $A, B \subseteq V(G)$ not necessarily disjoint. Then,*

$$e(A, B) \leq O(|A| + |B| + (|A||B|)^{(k+1)/(2k)}).$$

Proof. A well known fact (see e.g., [20]) in graph theory is that every m-edge graph has a bipartite subgraph with at least $m/2$ edges. Here we can apply the same idea. Let $C = A \cap B, A' = A \setminus C, B' = B \setminus C$. Form a partition of G as follows. Initialize $L = A', R = B'$

and then for every $c \in C$ place c in L with probability $1/2$ and in R otherwise. Then remove all edges not in $L \times R$. In expectation, at least half of the edges in $A \times B$ are preserved. Hence some setting of the random bits gives a bipartite subgraph with at least half the edges from $A \times B$ on which we can apply Theorem 1 to upper-bound $e(A,B)/2$ by $O(|L|+|R|+(|L||R|)^{(k+1)/(2k)})$ which is the desired quantity as $|L| \leq |A|$ and $|R| \leq |B|$. ◀

Now we prove a lemma to explain how consecutive layers in the decomposition of Lemma 5 interact.

▶ **Lemma 7.** *Let G be a C_{2k}-free graph with $\Delta(G) \leq m^{2/(k+1)}$. Fix $A \subseteq V(G)$ and $d \leq \Delta(G)$. Let B denote the set of vertices $v \in V(G)$ with $|N_A(v)| \in [d/2, d]$. Then,*

$$d\sqrt{|B|} \leq O(m^{\frac{1}{k+1}}\sqrt{|A|}).$$

Proof. Define $a = |A|, b = |B|, \Delta = \Delta(G)$. In this proof we will write $x \lesssim y$ to denote $x \leq O(y)$. We aim to show $d^2b/a \lesssim m^{2/(k+1)}$. This will follow from a simple chain of inequalities, with the main combinatorial ingredient being Corollary 6. In the inequalities we will use $\mathbb{1}_{[P]}$ to denote the indicator of whether P is true (i.e., it evaluates to 1 or 0 based on the truth of proposition P).

We have:

$$\begin{aligned}
d^2b/a &= \frac{(db)^2}{ab} \\
&\lesssim \min\left(\frac{d^2b}{a}, \frac{e(A,B)^2}{ab}\right) && B \text{ to } A \text{ degree regularity} \\
&\lesssim \min\left(\frac{d^2b}{a}, \frac{a^2+b^2+(ab)^{1+1/k}}{ab}\right) && \text{Corollary 6} \\
&\lesssim \min\left(\frac{d^2b}{a}, \frac{a}{b}\right) + \frac{b}{a} + \min\left(\frac{d^2b}{a}, (ab)^{1/k}\right) && \text{re-arrange terms} \\
&\lesssim \sqrt{\frac{d^2b}{a}\cdot\frac{a}{b}} + \frac{b}{a} + \min\left(\frac{d^2b}{a}, (ab)^{1/k}\right) && \min(x,y) \leq \sqrt{xy} \\
&\lesssim d + \frac{b}{a} + \min\left(\frac{d^2b}{a}, (ab)^{1/k}\right) && \text{simplify} \\
&\lesssim \Delta + \frac{b}{a} + \min\left(\frac{d^2b}{a}, (ab)^{1/k}\right) && d \leq \Delta \\
&\lesssim \Delta + \frac{\Delta\cdot a}{a} + \min\left(\frac{d^2b}{a}, (ab)^{1/k}\right) && b \leq \left|\bigcup_{v\in A} N(v)\right| \leq \Delta\cdot a \\
&\lesssim m^{2/(k+1)} + \min\left(\frac{d^2b}{a}, (ab)^{1/k}\right) && \Delta \leq m^{2/(k+1)} \\
&\lesssim m^{2/(k+1)} + \mathbb{1}_{[a>dm^{1-2/(k+1)}]}\cdot\frac{d^2b}{a} + \mathbb{1}_{[a\leq dm^{1-2/(k+1)}]}\cdot(ab)^{1/k} && \text{split min} \\
&\lesssim m^{2/(k+1)} + \frac{d^2b}{dm^{1-2/(k+1)}} + (dm^{1-2/(k+1)}b)^{1/k} && \text{monotonicity} \\
&\lesssim m^{2/(k+1)} + \frac{m}{m^{1-2/(k+1)}} + (m^{2-2/(k+1)})^{1/k} && db \leq O(m) \\
&\lesssim m^{2/(k+1)} && \text{simplify.}
\end{aligned}$$

◀

Now we combine Lemma 5 and Lemma 7 to obtain Theorem 4.

Proof of Theorem 4. We apply Lemma 5 to obtain $V_*, d_*, G', X_1, \ldots, X_k \subseteq V(G'), d_1, \ldots, d_k$ with the properties described in Lemma 5. We can easily count the number of k-walks in $X_1 \times \cdots \times X_k \times V(G')$ due to the regularity property that consecutive X_i's enjoy; it is at most

$$|X_1| \cdot \prod_{j=1}^{k} d_j. \tag{2}$$

We now bound Equation (2) using Lemma 7, which implies that for each $i \in [k-1]$,

$$d_i \sqrt{|X_i|} \leq O(m^{1/(k+1)} \sqrt{|X_{i+1}|}). \tag{3}$$

Repeatedly applying Equation (3) to Equation (2) gives:

$$|X_1| \prod_{j=1}^{k} d_j \leq \sqrt{|X_1||X_k|} d_k \cdot O(m^{(k-1)/(k+1)}). \tag{4}$$

To bound Equation (4) we make two observations. First, note that $|X_k| d_k \leq O(m)$ because all $v \in X_k$ have degree at least $d_k/2$. Second, recall that $X_1 \subseteq V_*$, and $d_* \geq d_k$. Thus, $|X_1| d_k \leq |X_1| d_* \leq O(m)$. Applying this in Equation (4) gives

$$|X_1| \prod_{j=1}^{k} d_j \leq O(m^{2k/(k+1)}). \tag{5}$$

By Lemma 5, the number of capped k-walks in G is at most $\widetilde{O}(1)$ times Equation (5); this is the desired bound. ◀

[8] showed that Theorem 4 implies a C_{2k} detection algorithm. The C_{2k} detection algorithm is based on the following slightly more general lemma; this lemma is also the basis of our listing algorithm.

▶ **Lemma 8.** *Fix parameter $\Delta \in \mathbb{N}$, and graph G. Let W be the number of capped k-walks in G that start at a vertex with degree at most Δ. There is a (randomized) algorithm to list the C_{2k}'s in G in time $\widetilde{O}(m^2/\Delta + W + t)$.*

Proof. Label the vertices $v_1, \ldots, v_n$, where $v_i \succ v_j$ iff $i > j$. For each $i \in [n]$, define a subgraph G_i on vertices $\{v_1, \ldots, v_i\}$, consisting of the edges that are part of the k-step BFS out of vertex v_i. For $i = 1, 2, \ldots, n$, we will list the C_{2k}'s in G_i that involve v_i. Note that this will result in listing all C_{2k}'s; in particular, if v_i is the largest (by $\succ$) vertex in some C_{2k}, then we will list this C_{2k} on iteration i of our procedure. Let t_i denote the number of C_{2k}'s in G_i that involve v_i; we now discuss how to list these C_{2k}'s. To list the C_{2k}'s in G_i involving v_i, we use color coding. We color each vertex with a random color from $[2k]$. Then we restrict our search to **colorful** C_{2k}'s: C_{2k}'s with a vertex of every color. Listing the colorful C_{2k}'s involving v_i can be done in time $O(|E(G_i)| + t_i)$. Each C_{2k} becomes colorful with probability $2^{-2k} \geq \Omega(1)$, so by repeating this $\widetilde{O}(1)$ times, we will list all the C_{2k}'s in G_i involving v_i, with high probability.

Now we analyze the cost of our algorithm. Note that $|E(G_i)|$ is at most the number of capped k-walks that start at v_i. Let T be the largest i such that $\deg(v_i) \leq \Delta$. Then, by definition

$$\sum_{i=1}^{T} |E(G_i)| \leq O(W).$$

There are at most m/Δ vertices of degree larger than Δ. Thus,

$$\sum_{i=T+1}^{n} |E(G_i)| \leq O(m^2/\Delta).$$

Finally, we have $\sum_i t_i = O(t)$. Combined, we see that the algorithm runs in time $\widetilde{O}(t + W + m^2/\Delta)$. ◀

▶ **Corollary 9.** *There is a (randomized) $\widetilde{O}(m^{2k/(k+1)})$ time algorithm for C_{2k} detection.*

Proof. This follows by applying Lemma 8 with $\Delta = m^{2/(k+1)}$ (and terminating the algorithm if it runs for too long). ◀

Lemma 8 and Corollary 9 can be derandomized using standard techniques (perfect hash families etc) [4].

3 Progress Towards the Supersaturation Conjecture

Jin and Zhou observed ([13]) that the unbalanced supersaturation conjecture implies sparse listing algorithms for even cycles of any length; we give a simple proof of this in Section A. In this section we present partial progress towards proving the unbalanced supersaturation conjecture. We restrict our attention to **hexagons** (C_6's) because this is the smallest case where sparse listing algorithms are not known. In Section 4 we will show that our partial progress towards the supersaturation conjecture can be translated into partial progress towards listing hexagons in time $O(m^{1.5} + t)$. Before stating our result, we state some simple folklore facts about path supersaturation (see Section B for proofs).

▶ **Fact 10.** Let G be a bipartite graph with parts A, B of sizes L, R. If $m \geq 2R$, then the number of 2-paths in $A \times B \times A$ is at least $L(m/L)(m/R)/2$.

▶ **Fact 11.** Let G be a bipartite graph with parts A, B of sizes L, R. If $m \geq 50(R+L)$, then the number of 4-paths in $A \times B \times A \times B \times A$ is at least $\Omega(L(m/L)^2(m/R)^2)$.

Our partial supersaturation result for hexagons is as follows:

▶ **Theorem 12.** *Let G be a bipartite graph with vertex bipartition $A \sqcup B$ and t hexagons. Let $L = |A|, R = |B|$ with $100 \leq L \leq R$, and let $d_L = m/L, d_R = m/R$. Suppose $m/\lceil \log(L+R) \rceil^{10} > L + R + (LR)^{2/3}$. Then, there is an absolute constant $c > 0$ such that*

$$t \geq d_R^3 d_L^3 \cdot \min(1, d_R^3/L, d_L^2 d_R^2/L^2) \cdot c/\log^{70} n.$$

Proof. Define $n = 2^{\lceil \log(L+R) \rceil}$. The constraint $100(\log n)^{10} \leq m \leq n^2$ implies $n \geq 10^8$.

Organizing the Graph. Our approach is to first organize the graph and then find hexagons via two and three step BFS's. The start of our organization of the graph is the following claim:

▷ Claim 13. There exist disjoint sets $A', A'' \subseteq A$ and $B', B'' \subseteq B$, of sizes $L' = |A'|, L'' = |A''|, R' = |B'|, R'' = |B''|$, and there exist values d'_L, d''_L, d'_R such that:

- Vertices $v \in A'$ have $|N_{B'}(v)| \in [d'_L/4, d'_L]$ and $e(A', B') \geq m/\log^4 n$.
- Vertices $v \in B'$ have $|N_{A''}(v)| \in [d'_R/4, d'_R]$ and $e(B', A'') \geq m/\log^4 n$.
- Vertices $v \in A''$ have $|N_{B''}(v)| \in [d''_L/4, d''_L]$ and $e(A'', B'') \geq m/\log^4 n$.

Proof. Define $Y_1 = Y_3 = A, Y_2 = Y_4 = B$. Form vertex subsets X_1, X_2, X_3, X_4 as follows. Define $X_4 = Y_4$. For $i = 3, 2, 1$ define X_i as follows: Partition $y \in Y_i$ based on which dyadic interval $|N_{X_{i+1}}(y)|$ falls in. Let X_i be a part in this partition with the maximum number of edges to X_{i+1}. The number of parts in our dyadic partition (disregarding the $\{0\}$ part, which does not contribute edges) is $\log n$. Thus,

$$e(X_4, X_3) \geq m/\log n, e(X_3, X_2) \geq m/\log^2 n, e(X_2, X_1) \geq m/\log^3 n. \tag{6}$$

Now, assign each vertex $v \in V(G)$ a uniformly random color from $\{0, 1\}$. For $i = 1, 2$ define X'_i to be the subset of X_i colored 0. For $i = 3, 4$ define X'_i to be the subset of X_i colored 1. Note that the sets X'_1, X'_2, X'_3, X'_4 are disjoint by construction: in particular, even though $X_2 \subseteq X_4$, by the coloring $X'_2 \cap X'_4 = \emptyset$.

We now argue that there exists some coloring such that the sets $\{X'_i \mid i \in [4]\}$ have similar sizes and degree regularity properties to the sets $\{X_i \mid i \in [4]\}$. First, because for each i, $|X_i| \geq 100$, with probability at least .9999 we have $|X'_i| \geq |X_i|/4$ for each i. Now, fix $i \in [3], v \in X_i$. The value $|N_{X'_{i+1}}(v)|$ is is a Bernoulli random variable with $|N_{X_{i+1}}(v)|$ trials, and $p = 1/2$ success probability for each trial. By a Chernoff bound,

$$\Pr[|N_{X'_{i+1}}(v)| \leq |N_{X_{i+1}}(v)|/4] \leq \exp(-|N_{X_{i+1}}(v)|/16). \tag{7}$$

Now, we will argue that $|N_{X_{i+1}}(v)|$ is large enough that we can afford to union bound across Equation (7) for all $i \in [3], v \in X_i$. For $i = 1, 2, 3$ choose d_i such that each vertex $v \in X_i$ has $|N_{X_{i+1}}(v)| \in [d_i/2, d_i]$; the sets X_i were constructed so that this is possible. Now, Equation (6) implies for $i \in [3]$ that

$$d_i \geq \frac{m}{|X_i| \log^3 n} \geq \log^7 n. \tag{8}$$

Therefore, the probability that Equation (7) fails for any $i \in [3], v \in X_i$ is at most

$$3n \exp(-(\log^7 n)/32) \leq .001.$$

Thus with probability at least .99, for each $i \in [4]$, $|X'_i| \geq |X_i|/4$, and if $i \neq 4$ then each vertex $v \in X'_i$ has $|N_{X'_{i+1}}(v)| \in [d_i/4, d_i]$, and finally (using Equation (6))

$$|X'_i| d_i \geq e(X'_i, X'_{i+1}) \geq \frac{1}{16} e(X_i, X_{i+1}) \geq m/\log^4 n.$$

Thus, by the probabilistic method there is some coloring that gives these properties. Fix this coloring, and let $A' = X'_1, A'' = X'_3, B' = X'_2, B'' = X'_4$. These are precisely the sets we wanted to show exist. ◁

We continue to use the sets A', B', A'', B'' introduced in Claim 13 throughout the proof, along with the values L', R', L'', R'' and d'_L, d''_L, d'_R. A useful immediate corollary of Claim 13 is

▶ **Corollary 14.** *$d'_L \geq (L/L')d_L/\log^4 n \geq d_L/\log^4 n$, $d'_R \geq (R/R')d_R/\log^4 n \geq d_R/\log^4 n$, and*
$d''_L \geq (L/L'')d_L/\log^4 n \geq d_L/\log^4 n$.

Now we further organize the vertices. In what follows, we will partition vertices based on the **approximate value** of some quantity associated with the vertex, by which we mean which of the dyadic intervals $\mathcal{I}_n$ the quantity lies in. For vertex v and vertex subset X, define $N^2_X(v)$ to be the set of vertices $w \in X$ that are distance 2 from v.

Fix vertex $v \in A'$. Define (X_v, x_v, δ_v) (visualized in Figure 2) as follows: Partition $w \in N^2_{A''}(v)$ into **buckets** based on the approximate value of $|N_{B'}(v) \cap N_{B'}(w)|$. Let $X_v \subseteq A''$ be a bucket which maximizes the number of edges between X_v and $N_{B'}(v)$. Let $x_v = |X_v|$, and choose $\delta_v \in \{1\} \cup \{4, 8, 16 \ldots, n\}$ such that each $w \in X_v$ has $|N_{B'}(v)| \in [\delta_v/2, \delta_v]$ (this is possible by the definition of $\mathcal{I}_n$, and the fact that the $\{0\}$ bucket does not contribute any edges). Then,

$$\delta_v x_v \geq e(X_v, N_{B'}(v)) \geq \frac{1}{\log n} e(N_{B'}(v), A'') \geq \frac{1}{16 \log n} \cdot d'_L d'_R. \tag{9}$$

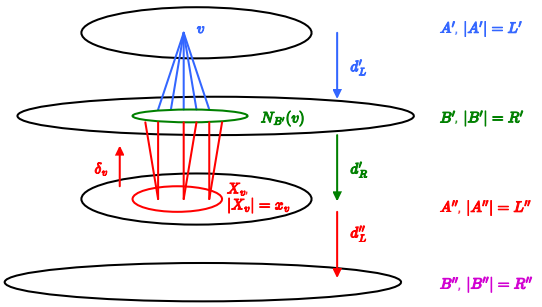


Figure 2 Organizing the graph.

Next, we assign each vertex $r \in N_{B'}(v)$ a tuple $(Y_{rv}, y_{rv}, \lambda_{rv})$ (visualized in Figure 3) as follows: Partition $z \in N^2_{B''}(r)$ into buckets based on the approximate value of $|N_{X_v}(r) \cap N_{X_v}(z)|$. Let $Y_{rv} \subseteq B''$ be a bucket which maximizes the number of edges between Y_{rv} and $N_{X_v}(r)$. Let $y_{rv} = |Y_{rv}|$ and choose $\lambda_{rv} \in \{1\} \cup \{4, 8, 16 \ldots, n\}$ such that each $z \in B''$ has $|N_{X_v}(r) \cap N_{X_v}(z)| \in [\lambda_{rv}/2, \lambda_{rv}]$. Then,

$$\lambda_{rv} y_{rv} \geq e(N_{X_v}(r), Y_{rv}) \geq \frac{1}{\log n} e(N_{X_v}(r), B''). \tag{10}$$

Finally, partition $r \in N_{B'}(v)$ into buckets based on the approximate values of $y_{rv}, \lambda_{rv}, |N_{X_v}(r)|$. Let $B'_v \subseteq B'$ be a bucket which maximizes the number of edges between B'_v and X_v. Choose y_v, λ_v, β_v (visualized in Figure 3) such that each $r \in B'_v$ has $y_{rv} \in [y_v/2, y_v]$, $\lambda_{rv} \in [\lambda_v/2, \lambda_v]$, and $|N_{X_v}(r)| \in [\beta_v/2, \beta_v]$. Then,

$$e(B'_v, X_v) \geq \frac{1}{\log^3 n} e(N_{B'}(v), X_v). \tag{11}$$

Counting Hexagons in the Organized Graph. Let $\mathsf{hex}(G)$ denote the number of hexagons in G. With the graph nicely organized we are prepared to bound $\mathsf{hex}(G)$. For vertex $v \in A'$, let $\mathsf{hex}(v)$ denote the number of hexagons in G where v is the only A' vertex participating in the hexagon. One natural way of bounding $\mathsf{hex}(G)$ is by $\mathsf{hex}(G) \geq \sum_{v \in A'} \mathsf{hex}(v)$. We will split into cases based on the characteristics of the 3-step BFS out of each vertex v and attempt to show that $\mathsf{hex}(v)$ is large. However, in one of the cases we will show $\mathsf{hex}(G)$ is large directly from the 3-step BFS out of a single vertex, rather than bounding $\mathsf{hex}(v)$. We now begin to consider 3-step BFS's based on their characteristics. See Figure 4 and Figure 5 for an illustration of our approach.

▷ Claim 15. Suppose vertex $v \in A'$ has $x_v d''_L/4 > 2R$ and $\delta_v \neq 1$. Then $\mathsf{hex}(v) \geq \Omega(d^3_R d^3_L/L')/\log^{26} n$.

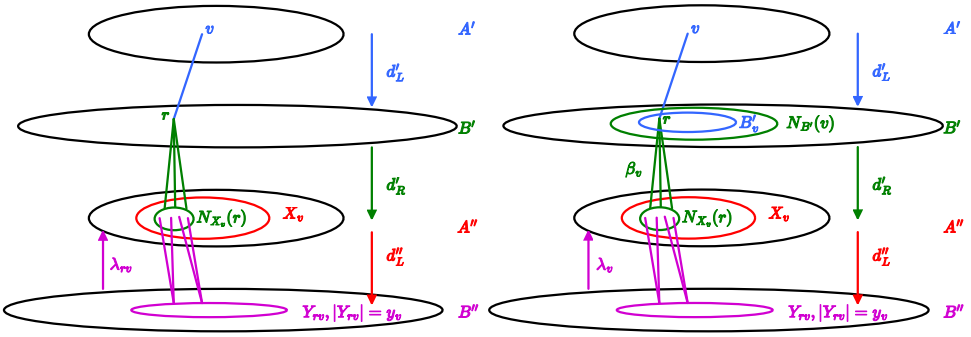


Figure 3 Organizing the graph.

Proof. Recall that $e(X_v, B'') \geq x_v d''_L/4$, which by assumption of the claim is at least $2R$. Thus, Fact 10 implies that the number of 2-paths in $X_v \times B'' \times X_v$ is at least

$$\Omega((x_v d''_L)^2/R). \tag{12}$$

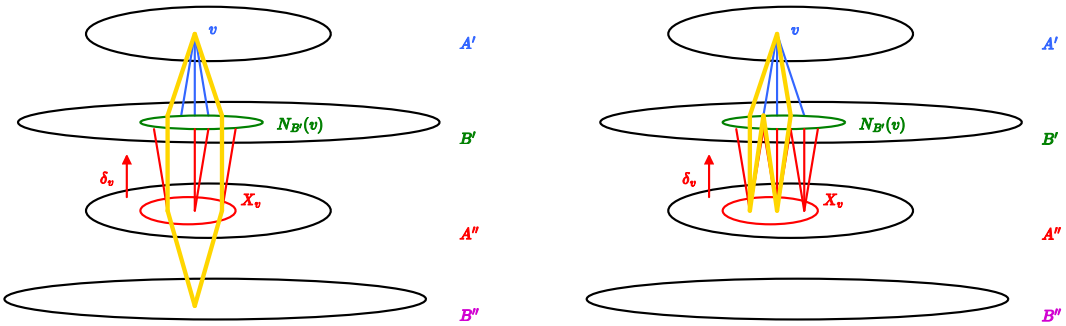
Given a two path $(w_1, z, w_2) \in X_v \times B'' \times X_v$, there are at least $(\delta_v/2)(\delta_v/2-1)$ ways to choose distinct $r_1 \in N_{B'}(w_1), r_2 \in N_{B'}(w_2)$. In this claim we are assuming $\delta_v \neq 1$ and consequentially $\delta_v \geq 4$. Thus, $(\delta_v/2)(\delta_v/2-1) \geq \delta_v^2/8$. Choosing r_1, r_2 in this manner, the vertices $(v, r_1, w_1, z, w_2, r_2)$ form a hexagon containing a single vertex from A'. Combined with Equation (12) this implies

$$\mathsf{hex}(v) \geq \Omega(\delta_v^2 (x_v d''_L)^2/R). \tag{13}$$

Using Equation (9) and Corollary 14 in Equation (13) gives

$$\mathsf{hex}(v) \geq \frac{1}{\log^2 n}\Omega((d'_L d'_R d''_L)^2/R) \geq \frac{1}{\log^{26} n}\Omega(d_R^3 d_L^3/L').$$

◁



(a) Claim 15. **(b)** Claim 19.

Figure 4 Two methods for showing that $\mathsf{hex}(v)$ is large.

Before proceeding to the next case, we make some simple observations.

▶ **Fact 16.** $d_R, d_L \geq \log^{10} n \geq 10^{14}$.

▶ **Corollary 17.** $d'_L d'_R d''_L/R \geq 10^{15} \log n$.

Proof. By Corollary 14 and Fact 16 we have

$$d'_L d''_L d'_R/R \geq (d_L^2 d_R/R)/\log^{12} n \geq (m^3/(LR)^2)/\log^{12} n \geq \log^{18} n \geq 10^{15}\log n. \quad \blacktriangleleft$$

▶ **Corollary 18.** $d'_R \geq 10^7 \log n$.

Proof. By Corollary 14 and Fact 16 we have

$$d'_R \geq d_R/\log^4 n \geq \log^6 n \geq 10^7 \log n. \quad \blacktriangleleft$$

▷ Claim 19. Suppose vertex $v \in A'$ has $x_v d''_L/4 \leq 2R$. Then $\mathsf{hex}(v) \geq \Omega((d_L d_R)^3 d_R^3/(LL'))/\log^{40} n$.

Proof. We claim that if x_v is this small then we have supersaturation of 4-paths between $N_{B'}(v)$ and X_v that start in $N_{B'}(v)$. Indeed, Equation (9), Corollary 17, Corollary 18 and our assumption that x_v is small imply

$$e(N_{B'}(v), X_v) \geq \frac{1}{16\log n} d'_L d'_R \geq 10^{12} R/d''_L + 10^5 d'_L \geq 50(d'_L + 8R/d''_L) \geq 50(d'_L + x_v).$$

Applying Fact 11, Equation (9), the hypothesis that x_v is small, and Corollary 14, the number of such 4-paths is at least

$$\frac{1}{\log^4 n}\Omega\left(\frac{(d'_L d'_R)^4}{d'_L x_v^2}\right) \geq \Omega((d_L d_R)^3 d_R^3/(LL'))/\log^{40} n.$$

Each of these 4-paths paired with v gives a hexagon that contributes to $\mathsf{hex}(v)$. This yields the desired bound on $\mathsf{hex}(v)$. ◁

The remaining two cases for what the BFS out of v looks like will both have $\delta_v = 1$. When $\delta_v = 1$, we have the following interesting property:

▷ Claim 20. Fix $v \in A'$ with $\delta_v = 1$. Then, $|B'_v| \geq d'_L/\log^5 n$ and $\beta_v \geq d'_R/\log^5 n$.

Proof. We count $e(B_{v'}, X_v)$ in two ways. First,

$$e(B_{v'}, X_v) \leq |B_{v'}|\beta_v. \tag{14}$$

Second, combining Equation (11) and Equation (9) we have

$$e(B_{v'}, X_v) \geq \frac{1}{\log^3 n} x_v \geq \frac{1}{16\log^4 n} d'_L d'_R. \tag{15}$$

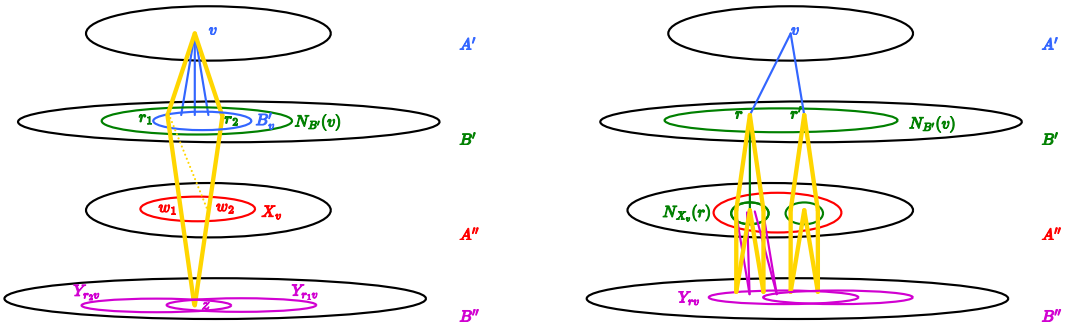
Combining Equation (14) and Equation (15) gives

$$|B'_v|\beta_v \geq \frac{1}{\log^5 n} d'_L d'_R. \tag{16}$$

Now we recall upper bounds for $|B_{v'}|, \beta_v$. Applying $\beta_v \leq d'_R$ in Equation (16) gives $|B'_v| \geq d'_L/\log^5 n$. Applying $|B'_v| \leq d'_L$ in Equation (16) gives $\beta_v \geq d'_R/\log^5 n$. ◁

We are now ready to handle the cases where $\delta_v = 1$.

▷ Claim 21. Fix $v \in A'$ and suppose $y_v d'_L > (4\log^5 n)R$ and $\delta_v = 1$. Then $\mathsf{hex}(v) \geq \Omega((d_L^3 d_R^3)/L')/\log^{46} n$.



(a) Claim 21. **(b)** Claim 22.

Figure 5 Handling $\delta_v = 1$.

Proof. We construct an auxiliary graph H between B'_v and B'' as follows: place an edge between $r \in B'_v$ and $z \in B''$ if r, z have a common neighbor in X_v. We have, $|E(H)| \geq |B'_v| y_v/2$. By Claim 20, and the hypothesis of the claim we have

$$|E(H)| \geq \frac{1}{2\log^5 n} d'_L y_v \geq 2R. \tag{17}$$

Thus, applying Fact 10 to Equation (17), the number of 2-paths in H starting in B'_v, is at least

$$\frac{1}{\log^{10} n} \Omega\left((d'_L y_v)^2/R.\right). \tag{18}$$

Now, fix some 2-path $(r_1, z, r_2) \in B'_v \times B'' \times B'_v$ in H. There are at least $\lceil \lambda_v/4 \rceil^2$ pairs $w_1, w_2 \in X_v$ such that $w_i \in N(z) \cap N(r_i)$ for $i = 1, 2$. We claim that for each such w_1, w_2, the tuple $(v, r_1, w_1, z, w_2, r_2)$ is a hexagon. This is not immediately obvious: we need to rule out the possibility of $w_1 = w_2$ (in which case we would have a 4-cycle with a dangling edge rather than a true hexagon). Fortunately, we can rule this out: the fact $\delta_v = 1$ precisely means that for $i = 1, 2$, w_i has only 1 neighbor in B'_v. Thus, we get at least $\lceil \lambda_v/4 \rceil^2$ times the quantity in Equation (18) many hexagons. That is,

$$\mathsf{hex}(v) \geq \frac{1}{\log^{10} n} \Omega((d'_L y_v \lambda_v)^2/R). \tag{19}$$

Now, recall that Equation (10) gives a bound on $y_v \lambda_v$: it asserts that for any $r \in B'_v$ we have

$$y_v \lambda_v \geq y_{rv} \lambda_{rv} \geq \frac{1}{\log n} e(N_{X_v}(r), B'') \geq \frac{1}{\log n} |N_{X_v}(r)| d''_L/4 \geq \frac{1}{\log n} (\beta_v/2) d''_L/4. \tag{20}$$

Now, using Claim 20 we have

$$y_v \lambda_v \geq \frac{1}{\log^6 n} \Omega(d'_R d''_L). \tag{21}$$

Now, we can bound Equation (19) by

$$\mathsf{hex}(v) \geq \frac{1}{\log^{22} n} \Omega((d'_L d'_R d''_L)^2)/R.$$

Now, using Corollary 14 we have

$$\begin{aligned}\mathsf{hex}(v) &\geq \frac{1}{\log^{22} n}\Omega((d'_L d'_R d''_L)^2)/R \\ &\geq \frac{1}{\log^{26} n}\Omega(d'_L(d'_R d''_L)^2(m/L')/R) \\ &\geq \frac{1}{\log^{46} n}\Omega(d_L(d_R d_L)^2(m/L')/R) \\ &= \frac{1}{\log^{46} n}\Omega(d_L^3 d_R^3/L').\end{aligned}$$

◁

The final case is slightly different. Instead of bounding $\mathsf{hex}(v)$ we will directly bound $\mathsf{hex}(G)$.

▷ Claim 22. Fix $v \in A'$ with $\delta_v = 1$, $y_v d'_L \leq (4\log^5 n)R$, and $x_v d''_L > 8R$. Then $\mathsf{hex}(G) \geq \Omega((d_L d_R)^5/L^2)/\log^{67} n$.

Proof. We will find hexagons of the following form: Take $r \in B'_v$, and then let $(x_1, x_2, x_3, x_4, x_5)$ be a 4-path between $N_{X_v}(r)$ and Y_{rv} (starting in $N_{X_v}(r)$). Then, $(r, x_1, x_2, x_3, x_4, x_5)$ is a hexagon. We claim that we have 4-path supersaturation between $N_{X_v}(r)$ and Y_{rv}. Indeed, by Equation (10) we have

$$e(N_{X_v}(r), Y_{rv}) \geq \frac{1}{\log n} e(N_{X_v}(r), B'') \geq \frac{1}{4\log n}|N_{X_v}(r)|d''_L \geq \frac{1}{8\log n}\beta_v d''_L.$$

Then, by Claim 20 we have

$$e(N_{X_v}(r), Y_{rv}) \geq \frac{1}{8\log^6 n} d'_R d''_L. \tag{22}$$

Now, recall that $|N_{X_v}(r)| \leq d'_R$, so Equation (22) together with Corollary 14 and Fact 16 imply $e(N_{X_v}, Y_{rv}) \geq 100|N_{X_v}(r)|$. Next, observe that by Corollary 14, Equation (22) implies

$$|Y_{rv}| \leq y_v \leq (4\log^5 n)R/d'_L \leq \frac{1}{4\log n}\frac{LR}{m} \leq \frac{1}{4\log^{31} n}\frac{m^2}{LR} \leq \frac{d_L d_R}{\log^{31} n} \leq \frac{d'_R d''_L}{800\log^6 n} \leq \frac{e(N_{X_v}(r), Y_{rv})}{100}.$$

Combined, we have

$$e(N_{X_v}(r), Y_{rv}) \geq 50(|Y_{rv}| + |N_{X_v}(r)|),$$

which is the required condition for 4-path supersaturation. Now, applying Fact 11 to Equation (22) the number of 4-paths that we get is:

$$\Omega\left(\frac{e(N_{X_v}(r), Y_{rv})^4}{|N_{X_v}(r)||Y_{rv}|^2}\right) \geq \frac{1}{\log^{24} n}\Omega\left(\frac{(d'_R d''_L)^4}{\beta_v y_v^2}\right) \tag{23}$$

Applying our assumption on y_v and the bound $\beta_v \leq d'_R$ gives

$$\frac{1}{\log^{34} n}\Omega\left(\frac{(d'_R d''_L)^4}{d'_R(R/d'_L)^2}\right) \tag{24}$$

Applying Corollary 14 to Equation (24), the number of such 4-paths is at least

$$\frac{m^9}{L^6 R^5}\Omega(1/\log^{61} n).$$

Each $r \in B'_v$ contributes this many hexagons, and the contributed hexagons are distinct. Applying Claim 20 we have

$$\mathsf{hex}(G) \geq |B'_v|\frac{m^9}{L^6 R^5}\Omega(1/\log^{61} n) \geq \frac{m^{10}}{L^7 R^5}\Omega(1/\log^{67} n) \geq (d_R d_L)^3(d_L d_R/L)^2/\log^{67} n.$$ ◁

Now, we can easily combine Claim 15, Claim 19, Claim 21, Claim 22 to deduce the theorem. Indeed, if any vertex $v \in A'$ falls into the case described by Claim 22 then have the desired bound on $\mathsf{hex}(G)$. Otherwise, each vertex must fall into one of the cases covered in Claim 15, Claim 19, Claim 21. Picking whichever case is most common of these, we can sum over the $\mathsf{hex}(v)$ bounds for at least $|A'|/3 = L'/3$ many $v \in A'$. This gives the desired bound on $\mathsf{hex}(G)$. ◀

4 Progress Towards Listing Algorithms

In this section we describe how our partial progress towards Conjecture 25 in Theorem 12 can be used together with our simplified analysis capped k-walks analysis from Section 2 to give improved listing algorithms for C_6's in sparse graphs. Our main result is:

▶ **Theorem 23.** *There is an algorithm for listing C_6's in time $\widetilde{O}(m^{1.6} + t)$.*

Theorem 23 follows immediately from Lemma 8 with $\Delta = m^{.4}$ and the following lemma:

▶ **Lemma 24.** *Fix G with $\Delta(G) \leq m^{.4}$. The number of capped 3-walks in G is at most $\widetilde{O}(m^{1.6} + t)$.*

Proof. If $n \leq 1000$ the result is trivial; we assume this is not the case. Fix $\lambda = 200\lceil \log n \rceil^{15}$. As in Theorem 4 and Theorem 28 we use Lemma 5 to obtain $\mathcal{B} = (V_*, d_*, G', X_1, X_2, X_3, d_1, d_2, d_3)$ with the properties described in Lemma 5. The number of 3-walks in $X_1 \times X_2 \times X_3 \times V(G')$ is at most $|X_1| \cdot d_1 d_2 d_3$; recall that the number of capped 3-walks in G is at most $\text{polylog}(n)|X_1|d_1d_2d_3$ – thus, in the rest of this proof we'll focus on bounding $|X_1|d_1d_2d_3$. At this point we break into 576 cases (with the help of a computer). A **case** is parameterized by a tuple $(B_{12}, B_{23}, D_{12}, D_{23}, U_{12}, U_{23}, A_{12}, A_{23})$, whose entries can take on the following values:

- $B_{12} \in \{1, 2\}, B_{23} \in \{2, 3\}$. B_{12} indicates which of X_1, X_2 is bigger (B stands for "bigger"). More precisely, $|X_{B_{12}}| = \max(|X_1|, |X_2|)$. B_{23} is defined analogously.
- $D_{12}, D_{23} \in \{1, 2, 3\}$. D_{12}, D_{23} indicate the "density regime" (D stands for "density") that the graphs between (X_1, X_2) and (X_2, X_3) fall into. This means, letting $R = \max(|X_1|, |X_2|), L = \min(|X_1|, |X_2|), m_{12} = e(X_1, X_2)$ and letting $v_1 = 1, v_2 = (d_R^3/L)/\lambda^3, v_3 = (d_L^2 d_R^2/L^2)/\lambda^4$, we have that $v_{D_{12}} = \min(v_1, v_2, v_3)$. D_{23} is defined analogously.
- $U_{12} \in \{0, 1\}, U_{23} \in \{0, 1\}$. U_{12} indicates whether the graph between X_1, X_2 is "balanced" or "unbalanced" (U stands for "unbalanced"). More precisely, U_{12} is 1 if $R \geq (RL)^{2/3}$ and 0 if $R \leq (RL)^{2/3}$. U_{23} is defined analogously.
- $A_{12}, A_{23} \in \{0, 1\}$. A_{12} indicates whether we can charge any hexagons to the graph between X_1, X_2 (A stands for "any"). More precisely, A_{12} is 1 if $m \geq \lambda \max(|X_1|, |X_2|, (|X_1||X_2|)^{2/3})$ and 0 if $m \leq \lambda \max(|X_1|, |X_2|, (|X_1||X_2|)^{2/3})$. A_{23} is defined analogously.

Note that $\mathcal{B}$ falls into at least one case (and the cases are slightly overlapping on the equality cases). Now, let $\mathcal{C} = (B_{12}, B_{23}, D_{12}, D_{23}, U_{12}, U_{23}, A_{12}, A_{23}) \in \{1, 2\} \times \{2, 3\} \times \{1, 2, 3\}^2 \times \{0, 1\}^4$ be a case that describes $\mathcal{B}$. For $i = 1, 2, 3$, let $x_i = \log_m(|X_i|), \delta_i = \log_m(d_i/\lambda)$. Let $\delta_* = \log_m(d_*/\lambda)$, and let τ denote $\log_m(t/c)$ where t is the number of C_6's in G, and c is the constant from Theorem 12. We now outline a series of *linear* constraints that $(x_1, x_2, x_3, \delta_1, \delta_2, \delta_3, \tau)$ must satisfy.

$$x_1, x_2, x_3 \in [0,1] \qquad \forall i, |X_i| \leq n \leq m.$$
$$\delta_*, \delta_1, \delta_2, \delta_3 \in [0, .4] \qquad \Delta(G) \leq m^{.4}.$$
$$\tau \geq 0$$
$$\forall i \in [3], \delta_i \leq \delta_*. \qquad \Delta(G') \leq \delta_*.$$
$$\delta_* + x_1 \leq 1, \qquad |X_1|\delta_* \leq 2m$$
$$\forall i \in [3], \delta_i + x_i \leq 1 \qquad \forall i, |X_i| d_i \leq 2m.$$

Next, based on B_{12}, B_{23} we get ordering relationships between x_1, x_2, x_3. For instance, if $B_{12} = 1, B_{23} = 3$ then we add the constraints

$$x_1 \geq x_2, x_2 \leq x_3.$$

We get more constraints from U_{12}, U_{23}. For instance, if $U_{12} = 0, U_{23} = 1$ we add the following constraints:

$$\max(x_1, x_2) \leq 2\min(x_1, x_2), \max(x_2, x_3) \geq 2\min(x_2, x_3).$$

Note that while these constraints appear non-linear (due to the $\max, \min$) they are actually linear, because B_{12}, B_{23} already specify orders between x_1, x_2, and between x_2, x_3! So, based on the value of B_{12}, B_{23} we could "unroll" these min / max's to the appropriate x_i; we decline to do so here for clarity.

Next, we implement constraints based on A_{12}, A_{23}. If $A_{12} = 1$ the constraint will be

$$x_1 + \delta_1 \geq \max(x_1, x_2, (2/3)(x_1 + x_2)).$$

The constraint would be flipped if $A_{12} = 0$. The constraint is analogous for A_{23}. Again, the max can be unrolled due to the value of the max being determined by B_{12}, U_{12}.

Next we implement constraints based on D_{12}, D_{23}. For instance, if $D_{12} = 1$ we would add constraints (where, as always, the $\max, \min$'s can be unrolled):

$$0 \leq 3(x_1+\delta_1) - 3\max(x_1, x_2) - \min(x_1, x_2), 0 \leq 4(x_1+\delta_1) - 2\max(x_1, x_2) - 4\min(x_1, x_2).$$

Collecting Hexagons. Now, with all these constraints in place, we will use Theorem 12 to obtain a lower bound on the number of hexagons in our graph. Then we'll set up an optimization problem that tells us the slowest possible time for our algorithm, observe that this optimization problem is an LP in each case, and solve the LPs on a computer to find a bound on the performance of our algorithm.

If $A_{12} = 1$, then letting $\ell = \min(x_1, x_2), r = \max(x_1, x_2), e = x_1 + \delta_1$ we add the constraint

$$\tau \geq 6e - 3(r + \ell) + \min(0, 3e - 3r - \ell, 4e - 4\ell - 2r).$$

This constraint is implied by Theorem 12, and we satisfy the pre-condition of Theorem 12 because $A_{12} = 1$. Note that as usual, the min / max's can be unrolled based on the case. If $A_{23} = 1$ we add an analogous constraint on τ.

Bounding capped-3 walks. Now, we want to answer the following question: "subject to $(x_1, x_2, x_3, \delta_*, \delta_1, \delta_2, \delta_3)$ satisfying the above constraints, is it possible that the number of 3-walks in $X_1 \times X_2 \times X_3 \times V(G')$ is larger than $\widetilde{O}(m^{1.6} + t)$? We can answer this question in

the negative by requiring the number of 3-walks to be more than $\Omega(t \log^{45} n)$, and showing that subject to this constraint, the number of 3-walks cannot be larger than $O(m^{1.6})$. In our linear programming formulation, this corresponds to adding the constraint

$$x_1 + \delta_1 + \delta_2 + \delta_3 \geq \tau,$$

and optimizing the following objective function, which is the number of 3-walks (up to $\operatorname{poly}\log n$ factors):

$$x_1 + \delta_1 + \delta_2 + \delta_3.$$

If the objective function optimizes to a number which is at most 1.6, then it will prove that the number of 3-walks of the specified form cannot be larger than $\widetilde{O}(m^{1.6} + t)$.

So, we can prove the lemma by iterating over the 576 possible cases, defining the appropriate linear program in each case, and checking that the optimum of the linear program is at most 1.6. We wrote a simple program [18] to iterate over the cases and solve the linear programs. Running the code we find that the optimum of each linear program is at most 1.6, as desired. We leave developing a human interpretable proof as an open question. ◀

References

1. Amir Abboud, Karl Bringmann, and Nick Fischer. Stronger 3-sum lower bounds for approximate distance oracles via additive combinatorics. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 391–404, 2023. doi:10.1145/3564246.3585240.
2. Amir Abboud, Karl Bringmann, Seri Khoury, and Or Zamir. Hardness of approximation in p via short cycle removal: cycle detection, distance oracles, and beyond. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1487–1500, 2022. doi:10.1145/3519935.3520066.
3. Amir Abboud, Seri Khoury, Oree Leibowitz, and Ron Safier. Listing 4-cycles. *arXiv preprint*, 2022. doi:10.48550/arXiv.2211.10022.
4. Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.
5. Noga Alon, Raphael Yuster, and Uri Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997. doi:10.1007/BF02523189.
6. John A Bondy and Miklós Simonovits. Cycles of even length in graphs. *Journal of Combinatorial Theory, Series B*, 16(2):97–105, 1974.
7. Karl Bringmann and Egor Gorbachev. A fine-grained classification of subquadratic patterns for subgraph listing and friends. *CoRR*, abs/2404.04369, 2024. doi:10.48550/arXiv.2404.04369.
8. Søren Dahlgaard, Mathias Bæk Tejs Knudsen, and Morten Stöckel. Finding even cycles faster via capped k-walks, 2017. arXiv:1703.10380.
9. Pierre-Louis Giscard, Paul Rochet, and Richard C Wilson. Evaluating balance on social networks from their simple cycles. *Journal of Complex Networks*, 5(5):750–775, 2017. doi:10.1093/COMNET/CNX005.
10. Tao Jiang and Liana Yepremyan. Supersaturation of even linear cycles in linear hypergraphs. *Combinatorics, Probability and Computing*, 29(5):698–721, 2020. doi:10.1017/S0963548320000206.
11. Ce Jin, Virginia Vassilevska Williams, and Renfei Zhou. Listing 6-cycles. In *Proceedings of the Symposium on Simplicity in Algorithms (SOSA 2024)*, pages 19–27, 2024. doi:10.1137/1.9781611977936.3.
12. Ce Jin and Yinzhan Xu. Removing additive structure in 3sum-based reductions. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 405–418, 2023. doi:10.1145/3564246.3585157.

13 Ce Jin and Renfei Zhou. Personal communication, 2024.

14 Steffen Klamt and Axel von Kamp. Computing paths and cycles in biological interaction graphs. *BMC bioinformatics*, 10:1–11, 2009. doi:10.1186/1471-2105-10-181.

15 Andrea Lincoln and Nikhil Vyas. Algorithms and Lower Bounds for Cycles and Walks: Small Space and Sparse Graphs. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2020.11.

16 V. Vassilevska. Efficient algorithms for path problems. *Ph.D. Thesis in Computer Science, Carnegie Mellon University*, 2008.

17 Virginia Vassilevska Williams and R. Ryan Williams. Subcubic equivalences between path, matrix, and triangle problems. *J. ACM*, 65(5), August 2018. doi:10.1145/3186893.

18 Alek Westover. Listing-c6s-lp. https://github.com/awestover/listing-C6s-LP/tree/main, 2024. Accessed: 2024-08-08.

19 Raphael Yuster and Uri Zwick. Finding even cycles even faster. *SIAM Journal on Discrete Mathematics*, 10(2):209–222, 1997. doi:10.1137/S0895480194274133.

20 Yufei Zhao. *Graph Theory and Additive Combinatorics: Exploring Structure and Randomness*. Cambridge University Press, 2023.

A Listing C_{2k}'s Using the Supersaturation Conjecture

Jin and Zhou [13] observed that the unbalanced supersaturation conjecture would allow one to obtain conditionally optimal sparse / dense listing algorithms for C_{2k}'s. In this section we give a simple proof of this observation, using the techniques of Section 2. Recall the unbalanced supersaturation conjecture:

▶ **Conjecture 25.** (The Unbalanced Supersaturation Conjecture [Jin and Zhou'24]) *Let G be an m edge bipartite graph with vertex bipartition $A \sqcup B$, with t C_{2k}'s. Suppose $m \geq 100k(|A| + |B| + (|A||B|)^{(k+1)/2k})$. Then,*

$$t \geq \Omega\left(\frac{m^{2k}}{|A|^k|B|^k}\right).$$

Conditional on Conjecture 25 we can follow the same proof strategy from Section 2 to obtain a C_{2k} listing algorithm. First, we need a simple consequence of Conjecture 25.

▶ **Corollary 26.** *Assume Conjecture 25. Fix G and $A, B \subseteq V(G)$* not necessarily disjoint. *Then,*

$$m \leq O(|A| + |B| + (|A||B|)^{(k+1)/(2k)} + t^{1/(2k)}\sqrt{|A||B|}).$$

Proof. As in Corollary 6, we pass to a bipartite subgraph with at least $m/2$ edges, while only decreasing $|A|, |B|, t$. Then we bound $m/2$ using Conjecture 25. ◀

Next we prove an analogue of Lemma 7, for analyzing the structure between two layers in the decomposition of Lemma 5.

▶ **Lemma 27.** *Assume Conjecture 25 Let graph G have $\Delta(G) \leq m^{2/(k+1)}$. Fix $A \subseteq V(G)$ and $d \leq \Delta(G)$. Let B denote the set of vertices $v \in V(G)$ that have $|N_A(v)| \in [d/2, d]$. Then,*

$$d\sqrt{|B|}/\sqrt{|A|} \leq O(m^{\frac{1}{k+1}} + t^{1/(2k)}).$$

Proof. The proof is nearly identical to that of Lemma 7. The only difference is that instead of applying the bound

$$e(A,B) \leq O(|A| + |B| + (|A||B|)^{(k+1)/(2k)}),$$

which applied when we assumed the graph was C_{2k}-free, we apply Corollary 26, which gives:

$$e(A,B) \leq O(|A| + |B| + (|A||B|)^{(k+1)/(2k)} + t^{1/(2k)}\sqrt{|A||B|}).$$

Substituting this new bound on $e(A,B)$ into the proof of Lemma 7 and otherwise leaving the analysis unchanged gives the desired bound on $d\sqrt{|B|/|A|}$. ◀

Now we deduce the analog of Theorem 4.

▶ **Theorem 28.** *Assume Conjecture 25. Let G be a graph with $\Delta(G) \leq m^{2/(k+1)}$. Then, the number of capped k-walks in G is at most $\widetilde{O}(m^{2k/(k+1)} + t)$.*

Proof. Just like in the proof of Theorem 4 we apply Lemma 5 to obtain V_*, d_*, G', $X_1, \ldots, X_k \subseteq V(G'), d_1, \ldots, d_k$ with the properties described in Lemma 5. The number of k-walks in $X_1 \times \cdots \times X_k \times V(G')$ is at most $|X_1| \cdot \prod_{j=1}^{k} d_j$. In Theorem 4 we bounded this product by repeatedly applying Lemma 7 (and then using an additional observation to bound $\sqrt{|X_1||X_k|}d_k$). Using Lemma 27 in place of Lemma 7 we have:

$$|X_1| \prod_{j=1}^{k} d_j \leq O((m^{1/(k+1)} + t^{1/(2k)})^{k-1} \cdot m). \tag{25}$$

Applying Hölder's Inequality to Equation (25) proves the desired bound on capped k-walks. ◀

Finally, we have the analog of Corollary 9.

▶ **Corollary 29.** *Assume Conjecture 25. Then, there is an algorithm for listing C_{2k}'s with running time $\widetilde{O}(t + \min(m^{2k/(k+1)}, n^2))$.*

Proof. First, we note that it is sufficient to establish the existence of an $\widetilde{O}(t + m^{2k/(k+1)})$ time algorithm. This is because balanced supersaturation results (Theorem 2) imply that if $m^{2k/(k+1)} > \Omega(n^2)$, meaning $m > \Omega(n^{1+1/k})$, then $t > \Omega(m^{2k/(k+1)})$, and so the t term dominates the running time. Now we give a listing algorithm with running time $\widetilde{O}(t + m^{2k/(k+1)})$.

Set $\Delta = m^{2/(k+1)}$. Assuming Conjecture 25, Theorem 28 implies that there are at most $\widetilde{O}(m^{2k/(k+1)} + t)$ capped-k walks starting at a vertex of degree at most Δ. Then, applying Lemma 8 gives a listing algorithm with the desired running time. ◀

B Folklore Results about Path Supersaturation

▶ **Fact 30.** Let G be a bipartite graph with parts A, B of sizes L, R. If $m \geq 2R$, then the number of 2-paths in $A \times B \times A$ is at least $L(m/L)(m/R)/2$.

Proof. Using Cauchy-Schwarz, the number of 2-paths is at least

$$\sum_{v \in B} \deg(v)(\deg(v) - 1) \geq \frac{1}{R}\left(\sum_{v \in B} \deg(v)\right)^2 - \sum_{v \in B} \deg(v) \geq m(m/R - 1) \geq m^2/(2R). \quad ◀$$

▶ **Fact 31.** Let G be a bipartite graph with parts A, B of sizes L, R. If $m \geq 50(R+L)$, then the number of 4-paths in $A \times B \times A \times B \times A$ is at least $\Omega(L(m/L)^2(m/R)^2)$.

Proof. Throughout the proof we use the notation $\deg_S(v) = |N(v) \cap S|$.

- Let B_1 be the set of vertices $v \in B$ with $\deg_A(v) \geq e(A,B)/(2|B|)$.
- Let A_1 be the set of vertices $v \in A$ with $\deg_{B_1}(v) \geq e(A,B_1)/(2|A|)$.
- Let B_2 be the set of vertices $v \in B_1$ with $\deg_{A_1}(v) \geq e(A_1,B_1)/(2|B_1|)$.

Observe that $e(A,B_1) \geq e(A,B)/2, e(A_1,B_1) \geq e(A,B)/4, e(A_1,B_2) \geq e(A,B)/8$. We now demonstrate that there are a large number of 4-paths starting in A. We can generate such a 4-path as follows:

- Fix an edge $(v_5, v_4) \in A_1 \times B_2$.
- Take $v_3 \in N_{A_1}(v_4)$, $v_2 \in N_{B_1}(v_3), v_1 \in N_A(v_2)$.

The number of such walks is at least:

$$\frac{e(A,B)}{8}\left(\frac{e(A,B)}{8|B_1|}-1\right)\left(\frac{e(A,B)}{4|A|}-1\right)\left(\frac{e(A,B)}{2|B|}-2\right) \geq \Omega(m^4/(LR^2)).$$ ◀

Polynomials, Divided Differences, and Codes

S. Venkitesh
Tel Aviv University, Israel

Abstract

Multiplicity codes (Kopparty et al., J. ACM 2014) are multivariate polynomial codes where the codewords are described by evaluations of polynomials (with a degree bound) and their derivatives up to some order (the multiplicity parameter), on a suitably chosen affine set of points. While efficient decoding algorithms were known in some special cases of point sets, by a reduction to univariate multiplicity codes, a general algorithm for list decoding up to the distance of the code when the point set is an arbitrary finite grid, was obtained only recently (Bhandari et al., IEEE TIT 2023). This required the characteristic of the field to be zero or larger than the degree bound, which is somewhat necessary since list decoding up to distance with small output list size is not possible when the characteristic is significantly smaller than the degree.

In this work, we present an alternative construction based on divided differences of polynomials, that conceptually resembles the classical multiplicity codes but has good list decodability "insensitive to the field characteristic". We obtain a simple algorithm that list decodes this code up to distance for arbitrary finite grids over all finite fields. Our construction can also be interpreted as a *folded Reed-Muller code*, which may be of independent interest.

2012 ACM Subject Classification Mathematics of computing → Coding theory; Theory of computation → Error-correcting codes

Keywords and phrases Error-correcting code, polynomial code, Reed-Solomon code, Reed-Muller code, folded Reed-Solomon code, folded Reed-Muller code, multiplicity code, divided difference, q-derivative, polynomial method, list decoding, list decoding capacity, linear algebraic list decoding

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.93

Related Version *Full Version*: `https://doi.org/10.48550/arXiv.2412.01755`

Funding A part of this work was done while the author was supported in part by BSF grant 2021683, ISF grant 735/20, and by the European Union (ERC, ECCC, 101076663).
S. Venkitesh: Len Blavatnik and the Blavatnik Family Foundation.

Acknowledgements The author thanks Dean Doron, Mrinal Kumar, Noga Ron-Zewi, Amnon Ta-Shma, and Mary Wootters for patiently listening to and giving feedback on presentations of this work at different times. A part of this work was done while the author was a postdoc at the Department of Computer Science, University of Haifa, and was participating in the program on Error Correcting Codes and Computation (2024) at the Simons Institute for the Theory of Computing, UC Berkeley.

1 Introduction and overview

Codes based on polynomial evaluations have found widespread applications, both implicitly and explicitly, in theoretical computer science, combinatorics, and allied areas. The prototypical univariate polynomial code is the *Reed-Solomon (RS) code* [57, 58], wherein the codewords are evaluations of degree bounded univariate polynomials on a set of points, and the prototypical multivariate polynomial code is the analogously defined *Reed-Muller (RM) code* [52, 57, 39, 68, 14]. [1] Despite several decades of research, the decodability properties of these codes are far from being well-understood.

[1] The RM code was considered over the binary field by [52, 57], and over larger fields by [39, 68, 14].

© S. Venkitesh;
licensed under Creative Commons License CC-BY 4.0

1.1 The list decoding problem

In this work, we are specifically interested in the *list decoding problem* [16, 69], where the aim is to output a list of close codewords for any given received word efficiently. The main parameter of interest is the *list decoding radius*, which represents the fraction of disagreements that we can decode from, and we would like it to be as large as possible.

Unique decoding of polynomial codes

To begin with, consider the easier setting where we fix the list decoding radius to be *half the minimum distance* of the code, which is precisely the setting of the *unique decoding problem* – in this case, there could only be at most one codeword this close to any given received word, and the aim is to efficiently find it if it exists. This problem has now been solved completely for RS codes [56, 49, 67], for univariate multiplicity codes [53], for RM codes in a special case [40, 5], and quite recently for RM codes in general [41] and for multivariate multiplicity codes [8]. What is noteworthy is that all of these algorithms are *insensitive to the field characteristic*. We now move on to consider larger list decoding radius.

List decoding of univariate polynomial codes

In the last decade of the 20th century, two classic works [65, 27] showed that all RS codes can be list decoded up to the *Johnson bound* (which is relative radius $1 - \sqrt{R}$, where R is the rate of the code), and also laid an algebraic algorithmic framework that has since been exploited over and over again for several other polynomial codes. Further, we do not know of explicit RS codes that can be list decoded beyond the Johnson bound, but we do know of explicit RS codes that are not list decodable significantly beyond the Johnson bound [7]. Due to recent breakthroughs [11, 26, 2], we now know that random RS codes are list decodable *up to (information theoretic) capacity* (that is, relative radius $1 - R - \epsilon$ for arbitrarily small but fixed $\epsilon > 0$, which represents getting approximately close to the information theoretic limit of list decoding), but we neither know of an explicit construction nor know of an algorithm to achieve this.

It is in this context of list decoding up to the information theoretic capacity that a *folding* trick has been enlightening. The works of [28, 29] showed that *folded Reed-Solomon (FRS) codes* (where the bits of an RS codeword are bunched into blocks of large constant size), and univariate multiplicity codes [43] (wherein the codewords are evaluations of polynomials and their derivatives up to a large constant multiplicity) can be algorithmically list decoded up to capacity! Further improvements to the parameters have been achieved in [44, 61, 63, 12] in different settings.

List decoding of multivariate polynomial codes

Moving on to multivariate polynomial codes, while the list decoding algorithms of [65, 27] generalize easily to RM codes, the decoding radius now worsens as a function of the number of variables, and attaining Johnson bound is no longer possible with these algorithms. While we know of a Johnson bound attaining algorithm [55] for RM codes when the finite grid is the full vector space $\mathbb{F}_q^m$ over the base field $\mathbb{F}_q$, designing a more general algorithm for arbitrary finite grids is still open!

As it turns out, the folding trick helps in the multivariate setting too. The multivariate multiplicity codes [45], which are the obviously defined multivariate analogues of univariate multiplicity codes, were shown to be list decodable *up to distance* (that is, up to radius $\delta - \epsilon$,

where δ is the minimum distance of the code) by [43] when the set of evaluation points is the full vector space $\mathbb{F}_q^m$. However, just as in the case of RM codes, an analogous result for arbitrary finite grids evaded us for quite some time. Very recently, [9] finally bolstered the algebraic algorithmic framework with some simple additional ingredients and managed to design an algorithm for list decoding multivariate multiplicity codes over arbitrary finite grids up to distance.

Sensitivity of list decodability to the field characteristic

Finally, yet another crucial component that could influence the performance of the code (specifically a polynomial code) is the most fundamental underlying algebraic object – the base field over which the polynomials are considered, which is always a finite field in our discussion.

Unlike the unique decoding setting that we mentioned earlier, we do observe sensitivity to the field characteristic in the setting of the list decoding problem. Let us fix the notation that $\mathbb{F}_q$ denotes the base field, and p denotes the characteristic of $\mathbb{F}_q$. On one hand, we know that when q is a large power of p, there are RS codes that cannot achieve anywhere close to list decoding capacity [7] since we can show instances of received words having superpolynomial sized output lists. On the other hand, upon large constant folding, the univariate multiplicity codes achieve capacity when p is larger than the degree parameter [29, 43], or smaller but linear in the degree [44], and the FRS codes achieve list decoding capacity [28, 29] insensitive to p. Thus, the list decodability of the FRS code is insensitive to the field characteristic, whereas the list decodability of the multiplicity code is sensitive to the field characteristic. Indeed, in the case when q is a large power of p, the large output lists shown by [7] for RS codes can be used to construct large output lists for multiplicity codes, and so the list decodability is provably poor.

Things get even more interesting in the multivariate setting, since the multivariate multiplicity codes achieve distance when p is larger than the degree parameter [55, 9], but we do not know of a field characteristic insensitive construction of a multivariate polynomial code that algorithmically achieves distance. Furthermore, we also do not know of an appropriate multivariate analogue of the field characteristic insensitive univariate FRS code. In this work, we answer both questions with a single code construction, along with a list decoding algorithm. In particular, our code subsumes the list decoding performance of the field characteristic sensitive multivariate multiplicity code [9], and also provides a natural *folded Reed-Muller code* construction. The motivation for our construction comes from an extremely simple yet foundational observation within the univariate world itself – the FRS code is also a multiplicity code!

1.2 FRS codes, univariate multiplicity codes, and divided differences

Fix a field $\mathbb{F}_q$. Consider distinct nonzero points $a_1, \ldots, a_n \in \mathbb{F}_q$. Let $\gamma \in \mathbb{F}_q^\times$ be a multiplicative generator, and suppose $s \geq 1$ such that the points $\gamma^j a_i$, $j \in [0, s-1]$, $i \in [n]$ are all distinct. For $k \in [sn]$, the degree-k Folded Reed-Solomon (FRS) code is defined by

$$\mathrm{FRS}_s(a_1, \ldots, a_n; k) = \left\{ [f]_{\mathrm{FRS}} := \begin{bmatrix} f(a_1) \\ f(\gamma a_1) \\ \vdots \\ f(\gamma^{s-1} a_1) \end{bmatrix} \middle| \cdots \middle| \begin{bmatrix} f(a_n) \\ f(\gamma a_n) \\ \vdots \\ f(\gamma^{s-1} a_n) \end{bmatrix} : \begin{array}{l} f(X) \in \mathbb{F}_q[X], \\ \deg(f) < k \end{array} \right\} \subseteq \left(\mathbb{F}_q^s\right)^n.$$

Here, the distance function is the Hamming distance on alphabet $\mathbb{F}_q^s$, that is, the Hamming weight [2] of a codeword $[f]_{\mathrm{FRS}}$ is the number of $i \in [n]$ such that the block $\begin{bmatrix} f(a_i) & f(\gamma a_i) & \cdots & f(\gamma^{s-1} a_i) \end{bmatrix}^{\mathsf{t}}$ is a nonzero vector in $\mathbb{F}_q^s$.

On the other hand, simply assuming $a_1, \ldots, a_n \in \mathbb{F}_q$ are distinct and nothing more about these points, for $k \in [sn]$, the degree-k (univariate) multiplicity code is defined by

$$\mathrm{Mult}_s(a_1, \ldots, a_n; k) = \left\{ [f]_{\mathrm{Mult}} := \begin{bmatrix} f^{(0)}(a_1) \\ f^{(1)}(a_1) \\ \vdots \\ f^{(s-1)}(a_1) \end{bmatrix} \middle| \cdots \middle| \begin{bmatrix} f^{(0)}(a_n) \\ f^{(1)}(a_n) \\ \vdots \\ f^{(s-1)}(a_n) \end{bmatrix} : \begin{array}{l} f(X) \in \mathbb{F}_q[X], \\ \deg(f) < k \end{array} \right\} \subseteq \left(\mathbb{F}_q^s\right)^n.$$

Here $f^{(j)}(X) := \frac{\mathrm{d}^j f(X)}{\mathrm{d}X^j}$ for all $j \in [0, s-1]$. [3] Once again, the Hamming weight of a codeword $[f]_{\mathrm{Mult}}$ is the number of $i \in [n]$ such that the block $\begin{bmatrix} f^{(0)}(a_i) & f^{(1)}(a_i) & \cdots & f^{(s-1)}(a_i) \end{bmatrix}^{\mathsf{t}}$ is a nonzero vector in $\mathbb{F}_q^s$.

Let us begin by looking at an algebraic feature that is intrinsic to multiplicity codes. Consider any $f(X) \in \mathbb{F}_q[X]$ with $\deg(f) < k$. For any $a \in \mathbb{F}_q$, we get the standard Taylor expansion

$$f(X) = f^{(0)}(a) + f^{(1)}(Y)(X-a) + f^{(2)}(a)\frac{(X-a)^2}{2!} + \cdots + f^{(k-1)}(a)\frac{(X-a)^{k-1}}{(k-1)!},$$

provided $p := \mathrm{char}(\mathbb{F}_q) \geq k$. Is there an analogous Taylor expansion in the context of the FRS code? As it turns out, there is such an expansion, and it is even simpler – it is the expansion in terms of the *Newton forward differences*. Precisely, we have

$$f(X) = f^{[0]}(a) + f^{[1]}(a)(X-a) + f^{[2]}(a)(X-a)(X-\gamma a) + \cdots + f^{[k-1]}(a)(X-a)\cdots(X-\gamma^{k-1}a),$$

where

$$f^{[0]}(a) = f(a), \;\; f^{[1]}(a) = \frac{f(\gamma a) - f(a)}{(\gamma-1)a}, \;\; \ldots, \;\; f^{[k-1]}(a) = \frac{f^{[k-2]}(\gamma a) - f^{[k-2]}(a)}{(\gamma^{k-1}-1)a}.$$

Notably, since $\gamma \in \mathbb{F}_q^\times$ is a multiplicative generator, we always have $\mathrm{ord}(\gamma) = q-1 \geq k$, and therefore, this expansion is valid *unconditional on* p.

In the first decade of the 20th century, Jackson [31, 32, 33] (also see [35, 34]) had defined the following specific *divided difference* operator on the polynomial ring,

$$\mathsf{D}_\gamma f(X) := \frac{f(\gamma X) - f(X)}{(\gamma-1)X}, \;\; \text{and} \;\; \mathsf{D}_\gamma^{t+1} := \mathsf{D}_\gamma \circ \mathsf{D}_\gamma^t \;\; \text{for all } t \geq 1.$$

In terms of this operator, the above expansion takes the form

$$f(X) = f(a) + \mathsf{D}_\gamma(a)(X-a) + \mathsf{D}_\gamma^2(a)\frac{(X-a)(X-\gamma a)}{[2]_\gamma!} + \cdots + \mathsf{D}_\gamma^{k-1}(a)\frac{(X-a)\cdots(X-\gamma^{k-1}a)}{[k-1]_\gamma!},$$

[2] We are only concerned with codes that are linear over the base field $\mathbb{F}_q$, and so the Hamming distance between two codewords c_1, c_2 is always equal to the Hamming weight of the codeword $c_1 - c_2$.

[3] Typically, since we are working over finite fields, there will also be a $j!$ scaling factor multiplied with the derivative, to prevent some pathologies arising in finite fields about higher multiplicity vanishing at a point. With this scaling factor, the derivative is usually called the Hasse derivative. We do not consider this, and implicitly assume that the characteristic is large enough so that these pathologies do not arise. Indeed, this is precisely the setting where the multiplicity codes have good list decodability.

where $[t]_\gamma := \frac{\gamma^t - 1}{\gamma - 1} = 1 + \gamma + \gamma^2 + \cdots + \gamma^{t-1}$, and $[t]_\gamma! := [t]_\gamma [t-1]_\gamma \cdots [1]_\gamma$. Returning to the code $\mathrm{FRS}_s(a_1, \ldots, a_n; k)$, it is then immediate by the definition of D_γ that there are invertible lower triangular matrices $U(a_i) \in \mathbb{F}_q^{s \times s}$, $i \in [n]$ such that for any codeword $[f]_{\mathrm{FRS}}$, we have

$$\begin{bmatrix} f(a_i) \\ \mathsf{D}_\gamma f(a_i) \\ \vdots \\ \mathsf{D}_\gamma^{s-1} f(a_i) \end{bmatrix} = U(a_i) \cdot \begin{bmatrix} f(a_i) \\ f(\gamma a_i) \\ \vdots \\ f(\gamma^{s-1} a_i) \end{bmatrix} \qquad \text{for all } i \in [n].$$

In other words, after a specific change of basis (see, for instance, [42, 4] for a proof), the FRS code can be interpreted as a *multiplicity code* with respect to the *derivative-like operator* D_γ.

Note that we have list decoding algorithms for FRS codes [28, 29], with the first one being algebraic and the second one being linear algebraic. If we are given the D_γ*-multiplicity encoding*, we may just change the basis to the usual FRS encoding and run one of these two FRS decoders. However, what is really encouraging is that we can straightforwardly adapt the linear algebraic multiplicity decoder [29] to our D_γ-multiplicity encoding, and the analysis will have a strong resemblance with that in the case of the classical derivatives. In this work, our main contribution is a construction that pushes this resemblance to the multivariate setting.

Our main algorithm will, in fact, work in the more general paradigm of list recovery, as was the case with the previous algorithms mentioned here.

Assumption on the field

Since our intention is to present the basic ideas as clearly and quickly as possible, we will stick to a simpler setting henceforth, where we assume that the evaluation points are contained in a finite field $\mathbb{F}_q$, but the polynomials are over a degree-3 field extension $\mathbb{K} = \mathbb{F}_{q^3}$. (See Remark 4.) We can relax this setting to some extent by tinkering with the parameters involved, but we will not make this effort.

The $\mathtt{Q}$-derivative

We will now forego the use of the symbol γ for the multiplicative generator. Instead we will denote $\mathtt{Q} \in \mathbb{K}^\times$ to be a multiplicative generator, and henceforth refer to the operator $\mathsf{D}_\mathtt{Q}$ as the $\mathtt{Q}$-derivative. This is the more standard terminology in the literature where this operator has appeared before. The $\mathtt{Q}$-derivative [4] is also called the *Jackson derivative*, and is a special case of the *Hahn derivative* [30], as well as the *Newton forward difference* [36, Chapter 1, Section 9]. See [37, Chapter 26] for a discussion on a slightly more general *quantum differential*, as well as some symmetrized variants. These are all instances of a broader notion of *divided difference* in *finite calculus* [46, 10, 54, 64, 36, 59]. The $\mathtt{Q}$-derivative seems to have been used so far only over fields of characteristic zero, particularly in *q-combinatorics* [18, 3, 21, 60, 20] and *quantum calculus* [37, 17, 47]. Further, the $\mathtt{Q}$-derivative does not seem to have made any explicit appearance so far in the *polynomial method* literature. However, as we see in this work, this notion turns out to be useful over fields of positive characteristic, and indeed, as part of the polynomial method.

[4] We use the notation '$\mathtt{Q}$' in $\mathtt{Q}$-derivative in this work instead of the more popular 'q', since we use q to denote the field size.

1.3 Multivariate $\mathtt{Q}$-derivatives, multivariate $\mathtt{Q}$-multiplicity code, and *folded RM code*

Our extension of $\mathtt{Q}$-derivatives from the univariate to multivariate setting is natural, and mimicks the extension of the classical derivatives. Assume indeterminates $\mathbb{X} = (X_1, \ldots, X_m)$, and for any $f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]$ and $\alpha \in \mathbb{N}^m$, we define the α-th $\mathtt{Q}$-derivative as the iterated operator

$$\mathsf{D}^{\alpha}_{\mathtt{Q}} f(\mathbb{X}) := \mathsf{D}^{\alpha_1}_{\mathtt{Q},X_1} \cdots \mathsf{D}^{\alpha_m}_{\mathtt{Q},X_m} f(\mathbb{X}),$$

where $\mathsf{D}_{\mathtt{Q},X_i}$ denotes the univariate $\mathtt{Q}$-derivative in the variable X_i. It is easy to see that the operator $\mathsf{D}^{\alpha}_{\mathtt{Q}}$ is well-defined and invariant under the order of iterations, just as in the case of the classical partial derivatives.

Now consider indeterminates $\mathbb{Y} = (Y_1, \ldots, Y_m)$. For any $t \geq 0$, denote $(X_i - Y_i)^{(t)}_{\mathtt{Q}} := (X_i - Y_i)(X_i - \mathtt{Q}Y_i) \cdots (X_i - \mathtt{Q}^{t-1}Y_i)$, and further, for any $\alpha \in \mathbb{N}^m$, denote $(\mathbb{X} - \mathbb{Y})^{(\alpha)}_{\mathtt{Q}} := \prod_{i=1}^{m} (X_i - Y_i)^{(\alpha_i)}_{\mathtt{Q}}$. Also denote $[\alpha]_{\mathtt{Q}}! := [\alpha_1]_{\mathtt{Q}}! \cdots [\alpha_m]_{\mathtt{Q}}!$, and $\mathtt{Q}^{\alpha}\mathbb{X} = (\mathtt{Q}^{\alpha_1}X_1, \ldots, \mathtt{Q}^{\alpha_m}X_m)$. Then for any $s \geq 1$ and $a \in (\mathbb{F}_q^{\times})^m$, there exists (see Proposition 9) an invertible matrix $U(a) \in \mathbb{K}^{\binom{m+s-1}{s-1} \times \binom{m+s-1}{s-1}}$ such that

$$\left[\mathsf{D}^{\gamma}_{\mathtt{Q}} f(a)\right]_{|\gamma|<s} = U(a) \cdot \left[f(\mathtt{Q}^{\gamma} a)\right]_{|\gamma|<s} \qquad \text{for all } f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]. \tag{1}$$

Fix any $s \geq 1$, and consider any nonempty $A \subseteq \mathbb{F}_q^{\times}$. For any $k \in [s|A|]$, we define the m-variate multiplicity-s degree-k $\mathtt{Q}$-multiplicity code by

$$\begin{aligned}&\mathtt{Q}\text{-Mult}_{m,s}(A;k)\\ &\quad = \left\{[\mathsf{D}_{\mathtt{Q}} \mid f] := \left[\left[\mathsf{D}^{\gamma}_{\mathtt{Q}} f(a)\right]_{|\gamma|<s}\right]_{a \in A^m} : f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}],\, \deg(f) < k\right\} \subseteq \left(\mathbb{K}^{\binom{m+s-1}{s-1}}\right)^{|A|^m}.\end{aligned}$$

Note that the distance function is now the Hamming distance on the alphabet $\mathbb{K}^{\binom{m+s-1}{s-1}}$. Further, by the change of basis that we just saw above, we can define a natural multivariate analogue of the univariate FRS codes. For any $k \in [s|A|]$, we define the s-folded degree k folded Reed-Muller (FRM) code by

$$\begin{aligned}&\mathrm{FRM}_{m,s}(A;k)\\ &\quad = \left\{[\mathtt{Q} \mid f] := \left[\left[f(\mathtt{Q}^{\gamma} a)\right]_{|\gamma|<s}\right]_{a \in A^m} : f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}],\, \deg(f) < k\right\} \subseteq \left(\mathbb{K}^{\binom{m+s-1}{s-1}}\right)^{|A|^m}.\end{aligned}$$

So by (1), we clearly have the change of basis

$$\mathtt{Q}\text{-Mult}_{m,s}(A;k) = \mathrm{diag}(U(a) : a \in A^m) \cdot \mathrm{FRM}_{m,s}(A;k).$$

The rate and distance of our code constructions can be given as follows, which will be immediate from Corollary 11 and Theorem 13.

▶ **Proposition 1.** *For any $k \in [s|A|]$, the codes $\mathtt{Q}$-Mult$_{m,s}(A;k)$ and $\mathrm{FRM}_{m,s}(A;k)$ have*

rate equal to $\dfrac{\binom{m+k-1}{k-1}}{\binom{m+s-1}{s-1}|A|^m}$, *and distance at least* $1 - \dfrac{k-1}{s|A|}$.

1.4 Our result

Our main result is that over *any* field $\mathbb{F}_q$ with sufficiently large q, and for sufficiently large constant multiplicity s, any constant rate multivariate $\mathtt{Q}$-multiplicity code $\mathtt{Q}$-Mult$_{m,s}(A;k)$, can be efficiently list decoded up to distance.

▶ **Theorem 2.** *Consider any $\delta \in (0,1)$, $\epsilon \in (0, 1-\delta)$. For any $A \subseteq \mathbb{F}_q^\times$ and for the choices $s = O(1/\epsilon^{2m+1})$ and $k = \lceil (1-\delta)s|A| \rceil$, the code $\mathtt{Q}$-$\mathrm{Mult}_s^m(A;k)$ is efficiently list decodable up to radius $\delta - \epsilon$ with output list contained in a $\mathbb{K}$-affine space of dimension atmost $O(1/\epsilon^{2m})$.*

Our list decoding algorithm is conceptually simpler than that by [9] for the classical multivariate multiplicity codes. Their algorithm in the classical setting entailed recovering the close enough polynomial messages one homogeneous component at a time, while using the classical *Euler's formula for homogeneous polynomials* (that relates a homogeneous polynomial to its first order derivatives) [5] to fully recover a homogeneous component (or a partial derivative of it) from its further first order partial derivatives. This formula, as well as the Taylor expansion for classical derivatives employed within the recovery step, both require the field characteristic to be large. In our $\mathtt{Q}$-setting, firstly there is no Euler's formula involving $\mathtt{Q}$-derivatives, and interestingly we don't need one! Secondly, the Taylor expansion for $\mathtt{Q}$-derivatives is clearly field characteristic insensitive. These are the only two places in the analysis where conceptually, the field characteristic was relevant in the classical setting, and is now irrelevant in the $\mathtt{Q}$-setting. Further, the short and clean analysis that we obtain encourages us to believe that this construction is the correct way to obtain a folded version of the RM code.

Let us give a quick outline of our algorithm. We make use of a clean *gluing* trick from [9] [6] and then proceed to perform a simpler analysis that is extremely close to the univariate analysis of [29]. The informal takeaway of what we show is that

$$\text{gluing trick} \;+\; \text{univariate list decoding analysis} \;=\; \text{multivariate list decoding analysis}.$$

Suppose we are given the code $\mathtt{Q}$-$\mathrm{Mult}_{m,s}(A;k)$ as in the statement of Theorem 2. Consider any received word

$$w = \Big(w_a := \Big[w_a^{(\gamma)}\Big]_{|\gamma|<s} : a \in A^m\Big) \in \big(\mathbb{K}^{\binom{m+s-1}{s-1}}\big)^{|A|^m}.$$

Consider two fresh sets of indeterminates $\mathbb{Y} = (Y_\gamma)_{|\gamma|<s}$ and $\mathbf{Z} = (Z_1, \ldots, Z_m)$. For a suitably chosen parameter $r \leq s$, we will find a nonzero interpolating polynomial with coefficients in $\mathbb{K}[\mathbf{Z}]$, having the form

$$P(\mathbb{X}, \mathbb{Y}, \mathbf{Z}) = \widetilde{P}(\mathbb{X}) + \sum_{j=0}^{r-1} P_j(\mathbb{X}) \Big(\sum_{|\gamma|=j} Y_\gamma \mathbf{Z}^\gamma \Big),$$

such that the following hold.

(Z1) The polynomials $\widetilde{P}(\mathbb{X}), P_0(\mathbb{X}), \ldots, P_{r-1}(\mathbb{X})$ must have suitably chosen $\mathbb{X}$-degree bounds. We will also simultaneously ensure a good $\mathbf{Z}$-degree bound on the coefficients of these polynomials by performing Gaussian elimination over $\mathbb{K}[\mathbf{Z}]$, as given in [38] or [9, Lemma 3].

(Z2) For any $f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]$, define a projection

$$P^{[f]}(\mathbb{X}) := \widetilde{P}(\mathbb{X}) + \sum_{j=0}^{r-1} \Big(\sum_{|\gamma|=j} \mathsf{D}_{\mathtt{Q}}^\gamma f(\mathbb{X}) \mathbf{Z}^\gamma \Big).$$

[5] Euler's formula states that for a homogeneous degree-d polynomial $f(\mathbb{X})$ over a field of characteristic greater than d, we have $\sum_{i=1}^m X_i \cdot \frac{\partial f(\mathbb{X})}{\partial X_i} = d \cdot f(\mathbb{X})$.

[6] As mentioned in [9], this gluing trick seems to have first appeared in the construction of a *hitting set generator* in [25].

Further, for every $\alpha \in \mathbb{N}^m$, $|\alpha| < s - r$, define a $\mathbb{K}(\mathbf{Z})$-linear operator $\Delta^{(\alpha)}$ (on an appropriate subspace of $\mathbb{K}(\mathbf{Z})[\mathbb{X}, \mathbb{Y}]$) that satisfies the condition $(\Delta^{(\alpha)}(P))^{[f]}(\mathbb{X}) = \mathsf{D}^{\alpha}_{\mathtt{Q}} P^{[f]}(\mathbb{X})$. The polynomial $P(\mathbb{X}, \mathbb{Y}, \mathbf{Z})$ must now satisfy the vanishing conditions

$$\Delta^{(\alpha)}(P)(a, w_a, \mathbf{Z}) = 0 \qquad \text{for all } \alpha \in \mathbb{N}^m \text{ with } |\alpha| < s - r, \text{ and } a \in A^m.$$

Note that the vanishing conditions in (Z2) define a linear system on the coefficients of $P(\mathbb{X}, \mathbb{Y}, \mathbf{Z})$, and we obtain the interpolating polynomial as a nontrivial solution of this linear system, by having large enough degree bounds as in (Z1). So for any close enough polynomial $f(\mathbb{X})$ (that is, having a large number of agreements with w), the conditions in (Z2) will imply that $P^{[f]}(\mathbb{X})$ vanishes at the agreement points with multiplicity at least $s - r$ (in the sense of multivariate $\mathtt{Q}$-derivatives, which we will define later), and this will imply that $P^{[f]}(\mathbb{X}) = 0$ by a suitable version of Polynomial Identity Lemma for multivariate $\mathtt{Q}$-derivatives. We can then solve this equation to recover $f(\mathbb{X})$. Importantly, the list decoding radius is dependent on the choice of r. We can achieve list decoding up to distance, that is, the claim of Theorem 2 by choosing $s = O(1/\epsilon^{2m})$ and $r = O(1/\epsilon^m)$. In fact, as in the analyses of [29, 9], we will show that the collection of all close enough polynomials is contained in a $\mathbb{K}$-affine subspace having dimension at most $\binom{m+r-2}{r-2}$.

This is a fairly simple strategy, very much within the algebraic algorithmic framework started employed in previous list decoding algorithms [65, 27, 28, 29, 43, 9]. Further, employing the gluing trick with the additional $\mathbf{Z}$-variables as in [9] allows us to formulate what we believe is the correct way to extend the linear algebraic decoding framework from the univariate to the multivariate setting, and our extension turns out to be simpler than the analysis in [9].

There is a subtlety in the recovery step in our multivariate setting vis-à-vis the univariate setting. When we solve the equation $P^{[f]}(\mathbb{X}) = 0$ to recover a close enough polynomial $f(\mathbb{X})$, we will see that the coefficients of $f(\mathbb{X})$ will be captured within the coefficients of some linear relations between monomials in the $\mathbf{Z}$-variables. Using a large enough efficient hitting set (an interpolating set of affine points that is large enough to certify all polynomials with a degree bound) in the $\mathbf{Z}$-variables, we can then recover each of these coefficients. This is similar to what was done in [9] in an analogous situation, but our overall analysis is a bit different and simpler. We will outline this difference now.

Formally, for a vector space $V_d \subseteq \mathbb{K}[\mathbf{Z}]$ of all polynomials having degree at most d, a hitting set is a finite set $H \subseteq \mathbb{K}^m$ such that for any $g(\mathbf{Z}) \in V_d$, if $g(\mathbf{Z}) \neq 0$ then there exists $u \in H$ such that $g(u) \neq 0$. For instance, by the Combinatorial Nullstellensatz [1, Theorem 1.1], every finite grid S^m with $S \subseteq \mathbb{K}$, $|S| > d$ is a hitting set for V_d. (We will precisely use this obvious hitting set in our analysis.) Our departure from the analysis of [9] is that they need to recover the coefficients of $f(\mathbb{X})$ one homogeneous component at a time, by employing the Euler's formula. In our analysis, we will directly recover one coefficient at a time (with respect to a suitable choice of basis), and this is extremely similar to the corresponding step in the linear algebraic list decoding algorithm [29] for classical univariate multiplicity codes. Recall that we intend to obtain an affine subspace of small dimension that contains all close enough polynomials. Assume the form $f(\mathbb{X}) = \sum_{|\alpha|<k} f_\alpha \mathbb{X}^\alpha$ for all *target polynomials* in the affine subspace. We will recover the target polynomials inductively from *lower coefficients to higher coefficients*. In order to kick-start the recovery procedure, as a base case of the induction, we set the coefficients $\big(f_\gamma : |\gamma| \leq r - 2\big)$ arbitrarily. Then for any $\alpha \in \mathbb{N}^m$, $|\alpha| \leq k - r$, the equation $P^{[f]}(\mathbb{X}) = 0$ would imply an equation that has the form

$$\begin{aligned}\mathbb{K}[\mathbf{Z}]\text{-linear combination}\Big(f_{\alpha+\gamma} : |\gamma| = r - 1\Big)& \\ = \ \mathbb{K}[\mathbf{Z}]\text{-linear combination}\Big(f_{\theta+\eta} : (|\eta|, \theta) \leq (r-1, \alpha),\ (|\eta|, \theta) \neq (r-1, \alpha)\Big),&\end{aligned}$$

where the polynomials comprising the above $\mathbb{K}[\mathbf{Z}]$-linear combinations all have a good $\mathbf{Z}$-degree bound due to the $\mathbf{Z}$-degree bound on $P(\mathbb{X}, \mathbb{Y}, \mathbf{Z})$. Note that each of the coefficients $f_{\theta+\eta}$ in the R.H.S. above is *strictly below* (in the usual componentwise partial order) some coefficient $f_{\alpha+\gamma}$ in the L.H.S. above. Since by induction hypothesis, we have already recovered all the coefficients in the R.H.S, we have the $\mathbb{K}[\mathbf{Z}]$-linear combination comprising the L.H.S. above also determined. We then instantiate a hitting set in the $\mathbf{Z}$-variables to recover the individual coefficients in the L.H.S. above, in much the same way as [9] instantiate for their linear combinations.

Thus, the coefficients $\left(f_\gamma : |\gamma| \leq r-2\right)$ are the only possible *free coefficients* in the above procedure, and therefore, the output list is contained in an affine $\mathbb{K}$-subspace having dimension at most $\binom{m+r-2}{r-2}$. We can further adapt the off-the-shelf *pruning* algorithm by [44] and its improved analysis by [66] to the multivariate setting, nearly identically as in [9], to finally end up with a randomized algorithm that guarantees constant output list size. Very recently, two concurrent works [63, 12] improved the output list size guarantee for list decoding FRS codes and univariate multiplicity codes from exponential in $1/\epsilon$ to $O(1/\epsilon^2)$ and the optimal $O(1/\epsilon)$ respectively. It is possible to extend their argument in the obvious way in the multivariate setting to give an even smaller output list size guarantee in the case of list decoding multivariate $\mathtt{Q}$-multiplicity codes. We do not mention the details in this presentation.

We will, in fact, prove a more general list recovery result as in Theorem 19, but stick to only showing polynomial output list size guarantee, for simplicity.

1.5 Further questions

Multivariate multiplicity codes evaluated on the vector space $\mathbb{F}_q^m$ are known to have good locality properties [45], especially when the field characteristic is larger than the degree [43, 44]. It will be interesting to see if similar locality properties hold for the multivariate $\mathtt{Q}$-multiplicity code in a field characteristic insensitive sense.

2 Preliminaries

List decoding and list recovery

Let Σ be a finite alphabet. An input list is any tuple $S = (S_1, \ldots, S_n)$ of subsets $S_i \subseteq \Sigma$. Further, for any $a \in \Sigma^n$, the (relative) Hamming distance between a and S is defined by

$$\mathsf{d}(a, S) = \frac{1}{n}|\{i \in [n] : a_i \notin S_i\}|.$$

We say an input list $S = (S_1, \ldots, S_n)$ is ℓ-sized if $|S_i| \leq \ell$ for all $i \in [n]$.

Let $\mathbb{F}$ be a field, and consider an $\mathbb{F}$-linear code C, that is, Σ is an $\mathbb{F}$-linear space.[7] For $\rho \in [0, 1]$ and $\ell, L \in \mathbb{Z}^+$, we say C is (ρ, ℓ, L)-list recoverable (LR) if there does not exist any ℓ-sized sequence of input lists S that admits $L+1$ distinct codewords $c^{(1)}, \ldots, c^{(L+1)} \in C$ such that

$$\mathsf{d}(c^{(j)}, S) \leq \rho, \qquad \text{for all } j \in [L+1].$$

The case $\ell = 1$ corresponds to list decoding with identical definitions.

[7] The more conventional definition defines C to be a linear code if C is linear over $\Sigma = \mathbb{F}$, but our definition is more relaxed where we allow Σ to be an $\mathbb{F}$-linear space. Our relaxed definition holds for our codes of interest, whereas the conventional definition does not hold. Indeed, we heavily use the relaxed linearity properties of our code constructions, often implicitly.

Field extensions

Fix a finite field $\mathbb{F}_q$, and a finite degree extension $\mathbb{K}/\mathbb{F}_q$ having extension degree κ, that is, $\mathbb{K} = \mathbb{F}_{q^\kappa}$. Also denote $[\kappa]_q = 1 + q + \cdots + q^{\kappa-1}$. Fix a multiplicative generator $\mathtt{Q} \in \mathbb{K}^\times$. We immediately note the following basic observations (see, for instance, [48, Chapter 2]), and also add quick proofs for completeness.

▶ **Proposition 3.**
(a) [48, Theorem 2.10] $\mathbb{K} = \mathbb{F}_q(\mathtt{Q})$.
(b) [48, Lemma 2.3] $\mathtt{Q}^t \notin \mathbb{F}_q$, *for all* $t \in [[\kappa]_q - 1]$.

Proof.
(a) Since $\mathbb{F}_q \subseteq \mathbb{K}$ and $\mathtt{Q} \in \mathbb{K}$, we have $\mathbb{F}_q(\mathtt{Q}) \subseteq \mathbb{K}$. Since $\mathbb{K}^\times = \{\mathtt{Q}^t : t \geq 0\}$, we have $\mathbb{K} \subseteq \mathbb{F}_q(\mathtt{Q})$.
(b) Since $\mathtt{Q}$ is the multiplicative generator of $\mathbb{K}^\times$, and $\mathbb{K} = \mathbb{F}_{q^\kappa}$, the multiplicative order of $\mathtt{Q}$ is $q^\kappa - 1$. Obviously, $\mathtt{Q}^t \neq 0$ for all $t \geq 1$. Now suppose $\mathtt{Q}^t \in \mathbb{F}_q^\times$ for some $t \geq 1$. Then $\mathtt{Q}^{t(q-1)} = 1$, which means $q^\kappa - 1 \leq t(q-1)$, that is, $t \geq (q^\kappa - 1)/(q-1) = 1 + q + \cdots + q^{\kappa-1} = [\kappa]_q$. ◀

▶ Remark 4. Note that in this work, we will assume that the field size $q \to \infty$. We will also consider an additional parameter $s \geq 1$, and always work in the case where $\mathtt{Q}, \ldots, \mathtt{Q}^{s-1} \notin \mathbb{F}_q$. By Proposition 3(b), this is true if $s - 1 < [\kappa]_q = 1 + q + q^2 + \cdots + q^{\kappa-1}$. Further, we will have the degree d of all polynomials that we consider to satisfy $d \leq sq - 1$. Therefore, we will also need $sq - 1 < [\kappa]_q = 1 + q + q^2 + \cdots + q^{\kappa-1}$. Both these requirements are satisfied if we take $\kappa = 3$ and $s \leq q$, and we assume these throughout the rest of this work. In fact, we will only take s to be a constant relative to q. So henceforth, we have $\mathbb{K} = \mathbb{F}_{q^3}$, and $\mathtt{Q}$ is a multiplicative generator of $(\mathbb{F}_{q^3})^\times$.

For any $n \geq k \geq 0$, denote

$$[n]_\mathtt{Q} = \sum_{t=0}^{n-1} \mathtt{Q}^t = \frac{\mathtt{Q}^n - 1}{\mathtt{Q} - 1}, \quad [n]_\mathtt{Q}! = \prod_{t=1}^{n} [t]_\mathtt{Q}, \quad \text{and} \quad \begin{bmatrix} n \\ k \end{bmatrix}_\mathtt{Q} = \frac{[n]_\mathtt{Q}!}{[k]_\mathtt{Q}![n-k]_\mathtt{Q}!}.$$

Note that $[n]_\mathtt{Q} \neq 0$ for all $n \in [q^\kappa - 2]$, and $[0]_\mathtt{Q}! = 1$. The quantity $\begin{bmatrix} n \\ k \end{bmatrix}_\mathtt{Q}$ is called the Gaussian binomial coefficient [22]. We will also have the convention that $[n]_\mathtt{Q}! = 0$ if $n < 0$, and $\begin{bmatrix} n \\ k \end{bmatrix}_\mathtt{Q} = 0$ if $k > n$.

2.1 Univariate $\mathtt{Q}$-derivatives

For any $f(X) \in \mathbb{K}[X]$, the $\mathtt{Q}$-derivative [31, 32, 33, 34, 35] is defined by

$$\mathsf{D}_\mathtt{Q} f(X) = \frac{f(\mathtt{Q}X) - f(X)}{(\mathtt{Q} - 1)X}.$$

Further, we are interested in iterated applications of the operator $\mathsf{D}_\mathtt{Q}$, and so we denote $\mathsf{D}_\mathtt{Q}^0 f(X) = f(X)$, and $\mathsf{D}_\mathtt{Q}^{t+1} f(X) = \mathsf{D}_\mathtt{Q}(\mathsf{D}_\mathtt{Q}^t f)(X)$ for all $t \geq 0$.

▶ Remark 5. We have $\mathsf{D}_\mathtt{Q}^t(X^k) = [k]_\mathtt{Q}[k-1]_\mathtt{Q} \cdots [k-t+1]_\mathtt{Q} X^{k-t+1}$ for all $k, t \geq 0$. Consider any nonconstant $f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]$ and $t \in [0, \deg(f)]$. Then we immediately get $\deg(\mathsf{D}_\mathtt{Q}^t f) \leq \deg(f) - t$ by linearity of $\mathsf{D}_\mathtt{Q}^t$ (see Proposition 6(a) below). More importantly, as a departure from classical derivatives, we get $\deg(\mathsf{D}_\mathtt{Q}^t f) = \deg(f) - t$ if $\deg(f) \leq [\kappa]_q - 1$.

Let us quickly collect a few basic properties of $\mathtt{Q}$-derivatives. For an indeterminate Y and $k \geq 0$, denote $(X - Y)_\mathtt{Q}^{(k)} = \prod_{t=0}^{k-1} (X - \mathtt{Q}^t Y)$. For any $f(X) \in \mathbb{K}[X]$ and $a \in \mathbb{K}^\times$, denote $(f \circ a)(X) = f(aX)$.

▶ **Proposition 6** ([37, 17]).

(a) Linearity. $\mathsf{D}_{\mathtt{Q}}^k$ *is an* $\mathbb{K}$*-linear map on* $\mathbb{K}[X]$*, for all* $k \geq 0$.

(b) Scaling. *For any* $f(X) \in \mathbb{K}[X]$ *and* $a \in \mathbb{K}^\times$,

$$\mathsf{D}_{\mathtt{Q}}^k(f \circ a)(X) = a^k \mathsf{D}_{\mathtt{Q}}^k f(aX) \qquad \text{for all } k \geq 0.$$

(c) Taylor expansion. *For any* $f(X) \in \mathbb{K}[X]$,

$$f(X) = \sum_{k \geq 0} \frac{\mathsf{D}_{\mathtt{Q}}^k f(Y)}{[k]_{\mathtt{Q}}!} (X - Y)_{\mathtt{Q}}^{(k)}.$$

In particular,

- *For any* $a \in \mathbb{K}$*, we have* $f(X) = \sum_{k \geq 0} \frac{\mathsf{D}_{\mathtt{Q}}^k f(a)}{[k]_{\mathtt{Q}}!} (X - a)_{\mathtt{Q}}^{(k)}$.
- *For any* $k \geq 0$*, we have* $\mathsf{D}_{\mathtt{Q}}^k f(0) = [k]_{\mathtt{Q}}! \cdot \mathsf{H}^{(k)} f(0)$*, where* $\mathsf{H}^{(k)} f(0)$ *is the* k-th Hasse derivative[8] *of* $f(X)$ *at* 0.

(d) Product rule. *For any* $f(X), g(X) \in \mathbb{K}[X]$ *and* $k \geq 0$,

$$\mathsf{D}_{\mathtt{Q}}^k(fg)(X) = \sum_{t=0}^{k} \begin{bmatrix} k \\ t \end{bmatrix}_{\mathtt{Q}} \mathsf{D}_{\mathtt{Q}}^{k-t} f(\mathtt{Q}^t X) \mathsf{D}_{\mathtt{Q}}^t g(X) = \sum_{t=0}^{k} \begin{bmatrix} k \\ t \end{bmatrix}_{\mathtt{Q}} \mathsf{D}_{\mathtt{Q}}^{k-t} f(X) \mathsf{D}_{\mathtt{Q}}^t g(\mathtt{Q}^{k-t} X).$$

Change of basis

A simple change of basis shows that $\mathtt{Q}$-derivatives are built out of simple evaluations at correlated points. Define an infinite matrix $\nu(X) \in (\mathbb{K}(X))^{\mathbb{N} \times \mathbb{N}}$ by

$$\nu_{k,t}(X) = \frac{(-1)^t \mathtt{Q}^{\binom{t}{2} - (k-1)t}}{(\mathtt{Q} - 1)^k X^k} \binom{k}{t}, \qquad \text{for all } k, t \geq 0.$$

Clearly, $\nu_{k,t}(X) = 0$ whenever $k < t$, that is, ν lower triangular. Further, we have the diagonal entries $\nu_{k,k}(X) = (-1)^k / (\mathtt{Q}^{\binom{k}{2}} (\mathtt{Q} - 1)^k X^k) \neq 0$ for all $k \geq 0$. So ν is invertible, and $\xi(X) := \nu(X)^{-1}$ is lower triangular, given by

$$\xi_{k,t}(X) = \begin{cases} (-1)^k \mathtt{Q}^{\binom{k}{2}} (\mathtt{Q} - 1)^k X^k & \text{if } k = t \geq 0, \\ \dfrac{(-1)^t (\mathtt{Q} - 1)^t X^t}{\mathtt{Q}^{\binom{k}{2} + (k-1)t}} \binom{k}{t} \left(1 - \left(1 - \dfrac{1}{\mathtt{Q}^{k-t-1}}\right)^{k-t}\right) & \text{if } k > t \geq 0. \end{cases}$$

This can be proven by instantiating, for instance, [62, Theorem 1.2.3].

▶ **Proposition 7** (Change of basis for $\mathtt{Q}$-derivatives). *For any* $f(X) \in \mathbb{K}[X]$ *and* $k \geq 0$*, we have*

$$\mathsf{D}_{\mathtt{Q}}^k f(X) = \sum_{t=0}^{k} \nu_{k,t}(X) f(\mathtt{Q}^t X) \quad [42, 4], \quad \text{and} \quad f(\mathtt{Q}^k X) = \sum_{t=0}^{k} \xi_{k,t}(X) \mathsf{D}_{\mathtt{Q}}^t f(X).$$

Proof. For any $f(X) \in \mathbb{K}[X]$, denote

$$(f \bullet \mathtt{Q}) = \left[f(\mathtt{Q}^k X)\right]_k \in (\mathbb{K}[X])^{\mathbb{N} \times 1} \quad \text{and} \quad (\mathsf{D}_{\mathtt{Q}} \bullet f) = \left[\mathsf{D}_{\mathtt{Q}}^k f(X)\right]_k \in (\mathbb{K}[X])^{\mathbb{N} \times 1}.$$

We have $(\mathsf{D}_{\mathtt{Q}} \bullet f) = \nu \cdot (f \bullet \mathtt{Q})$ by [42, 4]. Further, since $\xi = \nu^{-1}$, this implies $(f \bullet \mathtt{Q}) = \xi \cdot (\mathsf{D}_{\mathtt{Q}} \bullet f)$. ◀

[8] For $f(X) \in \mathbb{K}[X]$, the Hasse derivatives $\mathsf{H}^{(k)} f(Y)$, $k \geq 0$ satisfy: $f(X) = \sum_{k \geq 0} \mathsf{H}^{(k)} f(Y) (X - Y)^k$.

2.2 Multivariate $\mathtt{Q}$-derivatives

Now assume indeterminates $\mathbb{X} = (X_1, \ldots, X_m)$, and for any subset $I \subseteq [m]$, denote $\mathbb{X}_I = (X_i : i \in I)$. For every $i \in [m]$, denote the $\mathbb{K}$-linear $\mathtt{Q}$-derivative map on $\mathbb{K}[X_i]$ by $\mathsf{D}_{\mathtt{Q},X_i}$, and note that it extends in an obvious (and unique) way to a $\mathbb{K}(\mathbb{X}_{[m]\setminus\{i\}})$-linear map on $\mathbb{K}[\mathbb{X}]$. It is also immediate that $\mathsf{D}_{\mathtt{Q},X_1}, \ldots, \mathsf{D}_{\mathtt{Q},X_m}$ commute as $\mathbb{K}$-linear maps on $\mathbb{K}[\mathbb{X}]$. Then for any $f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]$ and $\alpha \in \mathbb{N}^m$, we define the α-th $\mathtt{Q}$-derivative by

$$\mathsf{D}^{\alpha}_{\mathtt{Q}} f(\mathbb{X}) = \mathsf{D}^{\alpha_1}_{\mathtt{Q},X_1} \cdots \mathsf{D}^{\alpha_m}_{\mathtt{Q},X_m} f(\mathbb{X}).$$

For $\alpha, \beta \in \mathbb{N}^m$, $\beta \le \alpha$, denote

$$[\alpha]_{\mathtt{Q}} = \prod_{i=1}^{m} [\alpha_i]_{\mathtt{Q}}, \quad [\alpha]_{\mathtt{Q}}! = \prod_{i=1}^{m} [\alpha_i]_{\mathtt{Q}}!, \quad \text{and} \quad \begin{bmatrix} \alpha \\ \beta \end{bmatrix}_{\mathtt{Q}} = \frac{[\alpha]_{\mathtt{Q}}!}{[\beta]_{\mathtt{Q}}![\alpha-\beta]_{\mathtt{Q}}!} = \prod_{i=1}^{m} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}_{\mathtt{Q}}.$$

Also, note the usual binomial coefficient $\binom{\alpha}{\beta} := \prod_{i=1}^{m} \binom{\alpha_i}{\beta_i}$. For indeterminates $\mathbb{Y} = (Y_1, \ldots, Y_m)$ and $\alpha \in \mathbb{N}^m$, denote $(\mathbb{X} - \mathbb{Y})^{(\alpha)}_{\mathtt{Q}} = \prod_{i=1}^{m} (X_i - Y_i)^{(\alpha_i)}_{\mathtt{Q}}$. For any $f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]$ and $a \in \mathbb{K}^{\times}$, denote $a\mathbb{X} = (a_1 X_1, \ldots, a_m X_m)$, and $(f \circ a)(\mathbb{X}) = f(a\mathbb{X})$. Also denote $a^{\gamma} = a_1^{\gamma_1} \cdots a_m^{\gamma_m}$. For any $\gamma \in \mathbb{N}^m$, denote $\mathtt{Q}^{\gamma}\mathbb{Y} = (\mathtt{Q}^{\gamma_1} Y_1, \ldots, \mathtt{Q}^{\gamma_m} Y_m)$.

We easily get the following basic properties of multivariate $\mathtt{Q}$-derivatives.

▶ **Proposition 8.**

(a) Linearity. $\mathsf{D}^{\alpha}_{\mathtt{Q}}$ *is a* $\mathbb{K}$*-linear map on* $\mathbb{K}[\mathbb{X}]$*, for all* $\alpha \in \mathbb{N}^m$*.*

(b) Scaling. *For any* $f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]$ *and* $a \in \mathbb{K}^{\times}$*, we have*

$$\mathsf{D}^{\beta}_{\mathtt{Q}}(f \circ a)(\mathbb{X}) = a^{\beta} \mathsf{D}^{\beta}_{\mathtt{Q}} f(a\mathbb{X}) \qquad \text{for all } \beta \in \mathbb{N}^m.$$

(c) Taylor expansion. *For any* $f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]$*,*

$$f(\mathbb{X}) = \sum_{\alpha \in \mathbb{N}^m} \frac{\mathsf{D}^{\alpha}_{\mathtt{Q}} f(\mathbb{Y})}{[\alpha]_{\mathtt{Q}}!} (\mathbb{X} - \mathbb{Y})^{(\alpha)}_{\mathtt{Q}}.$$

In particular,

- *For any* $a \in \mathbb{K}^m$*, we have* $f(X) = \sum_{\alpha \in \mathbb{N}^m} \frac{\mathsf{D}^{\alpha}_{\mathtt{Q}} f(a)}{[k]_{\mathtt{Q}}!} (\mathbb{X} - a)^{(\alpha)}_{\mathtt{Q}}$*.*
- *For any* $\alpha \in \mathbb{N}^m$*, we have* $\mathsf{D}^{\alpha}_{\mathtt{Q}} f(0^m) = [\alpha]_{\mathtt{Q}}! \cdot \mathsf{H}^{(\alpha)} f(0^m)$*, where* $\mathsf{H}^{(\alpha)} f(0^m)$ *is the* α-th Hasse derivative[9] *of* $f(\mathbb{X})$ *at* 0^m*.*

(d) Product rule. *For any* $f(\mathbb{X}), g(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]$ *and* $\alpha \in \mathbb{N}^m$*,*

$$\mathsf{D}^{\alpha}_{\mathtt{Q}}(fg)(\mathbb{X}) = \sum_{\beta \le \alpha} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}_{\mathtt{Q}} \mathsf{D}^{\alpha-\beta}_{\mathtt{Q}} f(\mathtt{Q}^{\beta}\mathbb{X}) \mathsf{D}^{\beta}_{\mathtt{Q}} g(\mathbb{X}) = \sum_{\beta \le \alpha} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}_{\mathtt{Q}} \mathsf{D}^{\alpha-\beta}_{\mathtt{Q}} f(\mathbb{X}) \mathsf{D}^{\beta}_{\mathtt{Q}} g(\mathtt{Q}^{\alpha-\beta}\mathbb{X}).$$

Proof. Each item follows easily by induction on m, with the base cases for Item (a), (b), (c), and (d) being Proposition 6(a), (b), (c), and (d) respectively. ◀

Change of basis

Using the infinite matrices defined in Section 2.1, define two infinite matrices $\nu^{(m)}(\mathbb{X}), \xi^{(m)}(\mathbb{X}) : (\mathbb{K}(\mathbb{X}))^{\mathbb{N}^m \times \mathbb{N}^m}$ by

$$\nu^{(m)}_{\alpha,\beta}(\mathbb{X}) = \prod_{i=1}^{m} \nu_{\alpha_i,\beta_i}(X_i), \quad \xi^{(m)}_{\alpha,\beta}(\mathbb{X}) = \prod_{i=1}^{m} \xi_{\alpha_i,\beta_i}(X_i), \qquad \text{for all } \alpha, \beta \in \mathbb{N}^m.$$

[9] For $f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]$, the Hasse derivatives $\mathsf{H}^{(\alpha)} f(\mathbb{Y})$, $\alpha \in \mathbb{N}^m$ satisfy: $f(\mathbb{X}) = \sum_{\alpha \in \mathbb{N}^m} \mathsf{H}^{(\alpha)} f(\mathbb{Y})(\mathbb{X} - \mathbb{Y})^{\alpha}$.

It is then easy to see (for instance, by induction on m) that $\nu^{(m)}(\mathbb{X})$ is invertible and $\xi^{(m)}(\mathbb{X}) = (\nu^{(m)}(\mathbb{X}))^{-1}$. Also, $\nu^{(m)}_{\alpha,\beta}(\mathbb{X}) = \xi^{(m)}_{\alpha,\beta}(\mathbb{X}) = 0$ for all $\alpha, \beta \in \mathbb{N}^m$, $\alpha \not\geq \beta$, and $\nu^{(m)}_{\alpha,\alpha}(\mathbb{X}) \neq 0 \neq \xi^{(m)}_{\alpha,\alpha}(\mathbb{X})$ as well as $\nu^{(m)}_{\alpha,\alpha}(\mathbb{X}) = 1/\xi^{(m)}_{\alpha,\alpha}(\mathbb{X})$ for all $\alpha \in \mathbb{N}^m$. Further, we obtain the following change of basis, again by induction on m.

▶ **Proposition 9** (Change of basis for multivariate $\mathtt{Q}$-derivatives). *For any $f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]$ and $\alpha \in \mathbb{N}^m$, we have*

$$\mathsf{D}^{\alpha}_{\mathtt{Q}} f(\mathbb{X}) = \sum_{\beta \leq \alpha} \nu^{(m)}_{\alpha,\beta}(\mathbb{X}) f(\mathtt{Q}^{\beta}\mathbb{X}), \quad \textit{and} \quad f(\mathtt{Q}^{\alpha}\mathbb{X}) = \sum_{\beta \leq \alpha} \xi^{(m)}_{\alpha,\beta}(\mathbb{X}) \mathsf{D}^{\beta}_{\mathtt{Q}} f(\mathbb{X}).$$

A monomial basis for function spaces

We assume familiarity with Gröbner basis theory. For definitions and more details, see [13, Chapters 2–5]. For any $A \subseteq \mathbb{F}_q^{\times}$, consider the ideal defined (by appealing to the change of basis in Proposition 9) as

$$\begin{aligned}\mathcal{D}^m_s(A) &= \{f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}] : f(\mathtt{Q}^{\gamma}a) = 0 \text{ for all } a \in A \text{ and } \gamma \in \mathbb{N}^m, |\gamma| < s\} \\ &= \{f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}] : \mathsf{D}^{\gamma}_{\mathtt{Q}} f(a) = 0 \text{ for all } a \in A \text{ and } \gamma \in \mathbb{N}^m, |\gamma| < s\}.\end{aligned}$$

The following is a characterization of a Gröbner basis of $\mathcal{D}^m_s(A)$ that is *universal* (invariant under choice of monomial order) and *reduced* (minimal with respect to divisibility of leading terms).[10] This is an extension of [1, Theorem 1.1] and [6, Theorem 4.1], while being a special case of [23, Theorem 4.7].[11] A similar characterization in the case of classical multivariate derivatives was given in [43, Section 6], and can also be obtained by [23, Theorem 4.7].

▶ **Theorem 10** ([23]). *For any $A \subseteq \mathbb{F}_q^{\times}$, the set of polynomials*

$$\mathcal{G}^m_s(A) := \left\{ \prod_{i=1}^{m} \prod_{t=0}^{\gamma_i - 1} \left(\prod_{a_i \in A} (X_i - \mathtt{Q}^t a_i) \right) : \gamma \in \mathbb{N}^m, |\gamma| = s \right\}$$

is a universal reduced Gröbner basis for $\mathcal{D}^m_s(A)$.

Further, for any $A \subseteq \mathbb{F}_q^{\times}$, define the function spaces

$$V_{m,s}(A) = \left\{ [f] := \left[[f]_a := \left[f(\mathtt{Q}^{\gamma}a) \right]_{|\gamma|<s} \right]_{a \in A^m} : f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}] \right\},$$

$$\text{and} \quad \mathsf{D}_{\mathtt{Q}} V_{m,s}(A) = \left\{ [\mathsf{D}_{\mathtt{Q}} \mid f] := \left[[\mathsf{D}_{\mathtt{Q}} \mid f]_a := \left[\mathsf{D}^{\gamma}_{\mathtt{Q}} f(a) \right]_{|\gamma|<s} \right]_{a \in A^m} : f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}] \right\}.$$

As an immediate corollary of Theorem 10 and the simple properties of Euclidean division for polynomials, we can describe a *monomial basis* (a basis where each vector is the appropriate evaluation of a monomial) for the above vector spaces.

[10] Given any monomial order, a universal Gröbner basis and a reduced Gröbner basis for an ideal both always exist; however, a Gröbner basis that is both universal and reduced may not always exist. A universal reduced Gröbner basis, whenever it exists, is the closest to being *the* Gröbner basis for an ideal, and allows performing Euclidean division *carelessly*.

[11] A related characterization of *standard monomials*, which is a strictly weaker notion than Gröbner basis, was obtained much earlier by [19, 50] via determination of winning strategies of a *lex game*, and a further reinterpretation of these winning strategies using the notion of *compression* for set systems. This reinterpretation was also rediscovered independently by [51].

▶ **Corollary 11.** *The collections of evaluation vectors*

$$\left\{[\mathbb{X}^\alpha] : \alpha \in \triangle^m_{s-1}(|A|)\right\} \quad \text{and} \quad \left\{[\mathsf{D}_\mathtt{Q} \mid \mathbb{X}^\alpha] : \alpha \in \triangle^m_{s-1}(|A|)\right\}$$

are $\mathbb{K}$-linear bases of $V_{m,s}(A)$ and $\mathsf{D}_\mathtt{Q}V_{m,s}(A)$ respectively, where

$$\triangle^m_{s-1}(|A|) = \left\{\alpha \in \mathbb{N}^m : \left\lfloor \frac{\alpha_1}{|A|} \right\rfloor + \cdots + \left\lfloor \frac{\alpha_m}{|A|} \right\rfloor \leq s-1 \right\}.$$

2.3 Counting Q-roots of polynomials with multiplicities

The discussion so far naturally leads us to the question of bounding the number of Q-*roots* of a polynomial with multiplicities.

For any nonzero univariate $f(X) \in \mathbb{K}[X]$ and $a \in \mathbb{K}$, define the Q-multiplicity of $f(X)$ at a by $\mu_\mathtt{Q}(f,a) = \min\left\{k \geq 0 : \mathsf{D}^k_\mathtt{Q} f(a) \neq 0\right\}$. The following claim is easy, by appealing to univariate factorization.

▶ **Proposition 12** (Easy). *For any nonempty set $A \subseteq \mathbb{F}_q^\times$, and nonzero $f(X) \in \mathbb{K}[X]$ with $\deg(f) \leq d < [3]_q$, we have $\sum_{a\in A} \mu_\mathtt{Q}(f,a) \leq d$. In particular, for any $s \geq 1$, we have $|\{a \in A : \mu_\mathtt{Q}(f,a) \geq s\}| \leq \lfloor d/s \rfloor$.*

In the multivariate setting, for any nonzero $f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]$ and $a \in \mathbb{K}^m$, define the Q-multiplicity of $f(\mathbb{X})$ at a by

$$\mu_\mathtt{Q}(f,a) = \min\left\{k \geq 0 : \text{there exists } \gamma \in \mathbb{N}^m,\ |\gamma| = k \text{ such that } \mathsf{D}^\gamma_\mathtt{Q} f(a) \neq 0\right\}.$$

In this section, we will prove the following multivariate extension of Proposition 12, which is analogous to [15, Lemma 2.7] in the setting of classical derivatives.

▶ **Theorem 13.** *For any nonempty set $A \subseteq \mathbb{F}_q^\times$, and any nonzero $f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]$ with $\deg(f) \leq d < [3]_q$, we have*

$$\sum_{a\in A^m} \mu_\mathtt{Q}(f,a) \leq d|A|^{m-1}.$$

In particular, for any $s \geq 1$, we have $|\{a \in A^m : \mu_\mathtt{Q}(f,a) \geq s\}| \leq \lfloor d\,|A|^{m-1}/s \rfloor$.

Towards a proof of Theorem 13, we first consider some simple consequences of the definition of multivariate Q-multiplicity.

▶ **Lemma 14.** *Consider any nonzero $f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]$.*

(a) *For any $a \in \mathbb{K}^m$, if $\mu_\mathtt{Q}(f,a) \geq k$, then*

$$\mu_\mathtt{Q}(\mathsf{D}^\gamma_\mathtt{Q} f, a) \geq k - |\gamma| \qquad \text{for all } \gamma \in \mathbb{N}^m,\ |\gamma| \leq k.$$

(b) *For any $(a_1,\ldots,a_m) \in \mathbb{K}^m$ and $\gamma \in \mathbb{N}^m$, if $\mathsf{D}^\gamma_\mathtt{Q} f(a_1,\ldots,a_{m-1},X_m) \neq 0$, then*

$$\mu_\mathtt{Q}(\mathsf{D}^\gamma_\mathtt{Q} f, (a_1,\ldots,a_{m-1},a_m)) \leq \mu_\mathtt{Q}(\mathsf{D}^\gamma_\mathtt{Q} f(a_1,\ldots,a_{m-1},X_m), a_m).$$

Proof.

(a) Since $\mu_\mathtt{Q}(f,a) \geq k$, we have

$$\mathsf{D}^\gamma_\mathtt{Q} f(a) = 0 \qquad \text{for all } \gamma \in \mathbb{N}^m,\ |\gamma| \leq k-1.$$

Consider any $\gamma, \beta \in \mathbb{N}^m$, $|\gamma| \leq k$, $|\beta| \leq k - |\gamma| - 1$. So $|\beta+\gamma| \leq k-1$, and therefore $\mathsf{D}^\beta_\mathtt{Q}(\mathsf{D}^\gamma_\mathtt{Q} f)(a) = \mathsf{D}^{\beta+\gamma}_\mathtt{Q} f(a) = 0$. This means $\mu_\mathtt{Q}(\mathsf{D}^\gamma_\mathtt{Q} f, a) \geq k - |\gamma|$.

(b) Suppose $\mu_{\mathtt{Q}}(\mathsf{D}^{\gamma}_{\mathtt{Q}} f, (a_1, \ldots, a_{m-1}, a_m)) = k$. This means

$$\mathsf{D}^{\beta+\gamma}_{\mathtt{Q}} f(a_1, \ldots, a_{m-1}, a_m) = 0 \qquad \text{for all } \beta \in \mathbb{N}^m,\ |\beta| \leq k-1.$$

In particular, restricting our attention to $\beta \in \mathbb{N}^m$ of the form $\beta = (0^{m-1}, t)$, we get

$$\mathsf{D}^{t}_{\mathtt{Q},X_m}(\mathsf{D}^{\gamma}_{\mathtt{Q}} f)(a_1, \ldots, a_{m-1}, a_m) = 0 \qquad \text{for all } t \leq k-1.$$

This means $\mu_{\mathtt{Q}}(\mathsf{D}^{\gamma}_{\mathtt{Q}} f(a_1, \ldots, a_{m-1}, X_m), a_m) \geq k$. ◀

We can now prove Theorem 13.

Proof of Theorem 13. We will prove by induction on m. The base case $m = 1$ is true by Proposition 12. Now suppose the assertion is true for some $m \geq 1$, and now consider indeterminates $\mathbb{X} = (X_1, \ldots, X_m, X_{m+1})$. Without loss of generality, assume $\deg(f) = d$, and write

$$f(X_1, \ldots, X_m, X_{m+1}) = \sum_{t=0}^{\ell} f_t(X_1, \ldots, X_m) \cdot X_{m+1}^t, \quad \text{where } f_\ell(X_1, \ldots, X_m) \neq 0,\ \deg(f_\ell) = d - \ell.$$

For any $a_1, \ldots, a_m \in A$, denote $s_{a_1,\ldots,a_m} = \mu_{\mathtt{Q}}(f_\ell, (a_1, \ldots, a_m))$. So by induction hypothesis, we have

$$\sum_{(a_1,\ldots,a_m)\in A^m} s_{a_1,\ldots,a_m} \leq (d-\ell)|A|^{m-1}. \tag{2}$$

Now fix $a_1, \ldots, a_m \in A$, and let $\gamma = (\gamma_1, \ldots, \gamma_m) \in \mathbb{N}^m$ such that $|\gamma| = s_{a_1,\ldots,a_m}$ and $\mathsf{D}^{\gamma}_{\mathtt{Q}} f_\ell(a_1, \ldots, a_m) \neq 0$. This means

- $\mathsf{D}^{\gamma}_{\mathtt{Q}} f_\ell(X_1, \ldots, X_m) \neq 0$, and so

$$\mathsf{D}^{(\gamma,0)}_{\mathtt{Q}} f(X_1, \ldots, X_m, X_{m+1}) = \sum_{t=0}^{\ell} \mathsf{D}^{\gamma}_{\mathtt{Q}} f_t(X_1, \ldots, X_m) \cdot X_{m+1}^t \neq 0.$$

- $g^{(\gamma)}(X_{m+1}) := \mathsf{D}^{(\gamma,0)}_{\mathtt{Q}} f(a_1, \ldots, a_m, X_{m+1}) \neq 0$, and $\deg(g^{(\gamma)}) = \ell$.

Then, for any $a_{m+1} \in A$, we get

$$\begin{aligned} \mu_{\mathtt{Q}}(f, (a_1, \ldots, a_m, a_{m+1})) &\leq |(\gamma, 0)| + \mu_{\mathtt{Q}}(\mathsf{D}^{(\gamma,0)}_{\mathtt{Q}} f, (a_1, \ldots, a_m, a_{m+1})) && \text{by Lemma 14(a)} \\ &\leq s_{a_1,\ldots,a_m} + \mu_{\mathtt{Q}}(g^{(\gamma)}, a_{m+1}) && \text{by Lemma 14(b)} \end{aligned}$$

So by Proposition 12,

$$\sum_{a_{m+1}\in A} \mu_{\mathtt{Q}}(f, (a_1, \ldots, a_m, a_{m+1})) \leq s_{a_1,\ldots,a_m}|A| + \ell.$$

Then (2) implies

$$\sum_{(a_1,\ldots,a_m,a_{m+1})\in A^{m+1}} \mu_{\mathtt{Q}}(f, (a_1, \ldots, a_m, a_{m+1})) \leq (d-\ell)|A|^m + \ell|A|^m = d|A|^m,$$

which completes the induction. ◀

3 List decoding/recovery of $\mathtt{Q}$-multiplicity codes

We will now move to our main result on list decoding multivariate $\mathtt{Q}$-multiplicity codes. In fact, we will present the algorithm for the more general list recovery paradigm.

3.1 List decoding/recovery of univariate $\mathtt{Q}$-multiplicity codes

Let us see that the classical univariate multiplicity code list decoder [29] can be suitably adapted to list decode $\mathtt{Q}$-Mult$_s(A;k)$, which therefore provides an alternative algorithm to list decode FRS codes. We will, in fact, consider list recovery.

▶ **Theorem 15.** *Consider any $R \in (0,1)$, $\epsilon \in (0, 1-R)$. For any $A \subseteq \mathbb{F}_q^\times$, $|A| = n$ and $\ell \geq 1$, and for the choices $s = \lceil \ell/\epsilon^2 \rceil$ and $k = \lceil Rsn \rceil$, the code $\mathtt{Q}$-Mult$_s(A;k)$ is efficiently list recoverable up to radius $1 - R - \epsilon$ with output list size at most $(\ell/\epsilon)^{O((1+\log \ell)/\epsilon)}$.*

Consider additional indeterminates $\mathbb{Y} = (Y_0, \ldots, Y_{s-1})$, and the $\mathbb{K}$-linear subspace of the polynomial ring $\mathbb{K}[X, \mathbb{Y}]$ defined by

$$\mathsf{L}_s(X, \mathbb{Y}) = \big\{ \widetilde{P}(X) + P_0(X)Y_0 + \cdots + P_{s-1}(X)Y_{s-1} : \widetilde{P}(X), P_0(X), \ldots, P_{s-1}(X) \in \mathbb{K}[X] \big\}.$$

Also, denote $Y_s = 0$. Define a $\mathbb{K}$-linear operator $\Delta : \mathsf{L}_s(X, \mathbb{Y}) \to \mathsf{L}_s(X, \mathbb{Y})$ by

$$\Delta\bigg(\widetilde{P}(X) + \sum_{j=0}^{s-1} P_j(X) Y_j \bigg) = \mathsf{D}_\mathtt{Q} \widetilde{P}(X) + \sum_{j=0}^{s-1} \big(\mathsf{D}_\mathtt{Q} P_j(X) Y_j + P_j(\mathtt{Q}X) Y_{j+1} \big).$$

We will be interested in iterated applications of Δ, and therefore, denote $\Delta^0 = \mathrm{Id}$ and $\Delta^{r+1} = \Delta \circ \Delta^r$ for all $r \geq 0$. For any $P(X, \mathbb{Y}) \in \mathsf{L}_s(X, \mathbb{Y})$ and $f(X) \in \mathbb{K}[X]$, denote $P^{[f]}(X) = P(X, f(X), \mathsf{D}_\mathtt{Q} f(X), \ldots, \mathsf{D}_\mathtt{Q}^{s-1} f(X))$. The following is immediate by the definition of Δ.

▶ **Lemma 16.** *Let $P(X, \mathbb{Y}) \in \mathsf{L}_s(X, \mathbb{Y})$.*

(a) *If $j \in [0, s-1]$ such that $\deg_{Y_{j'}}(P) = 0$ for all $j' \geq j$, then $\deg_{Y_{j'}}(\Delta P) = 0$ for all $j' \geq j+1$.*

(b) *$(\Delta P)^{[f]}(X) = \mathsf{D}_\mathtt{Q}(P^{[f]})(X)$, for all $f(X) \in \mathbb{K}[X]$.*

Proof. Let $P(X, \mathbb{Y}) = \widetilde{P}(X) + P_0(X)Y_0 + \cdots + P_{s-1}(X)Y_{s-1} \in \mathsf{L}_s(X, \mathbb{Y})$.

(a) If $j \in [0, s-1]$ such that $\deg_{Y_j}(P) = 0$ for all $j' \geq j$, this means $P_j(X) = \cdots = P_{s-1}(X) = 0$. Now we have

$$\begin{aligned}(\Delta P)(X, \mathbb{Y}) &= \mathsf{D}_\mathtt{Q} \widetilde{P}(X) + \sum_{j'=0}^{s-1} \big(\mathsf{D}_\mathtt{Q} P_{j'}(X) Y_j + P_{j'}(\mathtt{Q}X) Y_{j'+1} \big) \\ &= \mathsf{D}_\mathtt{Q} \widetilde{P}(X) + \sum_{j'=0}^{s-1} \big(P_{j'-1}(\mathtt{Q}X) + \mathsf{D}_\mathtt{Q} P_{j'}(X) \big) Y_{j'}.\end{aligned}$$

So, $P_{j'-1}(\mathtt{Q}X) + \mathsf{D}_\mathtt{Q} P_{j'}(X) = 0$ for all $j' \geq j+1$. This means $\deg_{Y_{j'}}(\Delta P) = 0$ for all $j' \geq j+1$.

(b) We have

$$\begin{aligned}(\Delta P)^{[f]}(X) &= \mathsf{D}_\mathtt{Q} \widetilde{P}(X) + \sum_{j=0}^{s-1} \big(\mathsf{D}_\mathtt{Q} P_j(X) \mathsf{D}_\mathtt{Q}^j f(X) + P_j(\mathtt{Q}X) \mathsf{D}_\mathtt{Q}^{j+1} f(X) \big) \\ &= \mathsf{D}_\mathtt{Q} \widetilde{P}(X) + \sum_{j=0}^{s-1} \mathsf{D}_\mathtt{Q} \big(P_j \cdot \mathsf{D}_\mathtt{Q}^j f \big)(X) \qquad \text{by Proposition 6(c)} \\ &= \mathsf{D}_\mathtt{Q} P^{[f]}(X).\end{aligned}$$

◀

It is worth noting that a close variant of the operator Δ appears in [24, Definition 8.1] in the context of nearly linear-time list decoding of FRS codes, without the interpretation in terms of $\mathtt{Q}$-derivatives. So it is reasonable to surmise that a nearly linear-time implementation of the list decoding algorithm à la [24] is possible for the univariate $\mathtt{Q}$-multiplicity codes. In order to keep our main presentation short, we limit ourselves here to adapting the more conventional polynomial-time algorithm of [29].

The interpolation step of the list recovery algorithm, which will be used multiple times in this discussion, can be captured as follows.

▶ **Proposition 17.** *Consider any $A = \{\alpha_1, \ldots, \alpha_n\} \subseteq \mathbb{F}_q^\times$, parameters $s \geq r \geq 1$, and $k \in [s|A|]$. Let $\ell \geq 1$, and define*

$$d = \left\lceil \frac{n\ell(s-r+1) - (r+k) + 1}{r+1} \right\rceil.$$

Then for any $S_1, \ldots, S_n \subseteq \mathbb{K}^s$ with $|S_i| \leq \ell$, $i \in [n]$, there exists a nonzero polynomial $P(X, \mathbb{Y}) = \widetilde{P}(X) + \sum_{j=0}^{r-1} P_j(X) Y_j \in \mathsf{L}_s(X, \mathbb{Y})$ with

$$\deg(\widetilde{P}) \leq d + k - 1, \quad \textit{and} \quad \deg(P_j) \leq d \qquad \textit{for all } j \in [0, r-1],$$

such that for any $f(X) \in \mathbb{K}[X]$, $\deg(f) < k$,

$$\textit{if} \quad \mathsf{d}_A^{(s)}(f, S) \geq \frac{\ell}{r+1} + \frac{k}{(s-r+1)n} \qquad \textit{then} \quad P(X, f(X), \mathsf{D}_{\mathtt{Q}} f(X), \ldots, \mathsf{D}_{\mathtt{Q}}^{s-1} f(X)) = 0.$$

Proof. Assume the notation $u = (u^{(0)}, \ldots, u^{(s-1)}) \in \mathbb{K}^s$. We can, in fact, add additional arbitrary elements and assume $|S_i| = \ell$, for all $i \in [n]$. Also enumerate $S_i = \{w_{i,1}, \ldots, w_{i,\ell}\}$, $i \in [n]$. We will construct a nonzero polynomial $P(X, \mathbb{Y}) = \widetilde{P}(X) + \sum_{j=0}^{s-1} P_j(X) Y_j \in \mathsf{L}_s(X, \mathbb{Y})$ such that

(a) $P(X, \mathbb{Y}) = \widetilde{P}(X) + \sum_{j=0}^{r-1} P_j(X) Y_j \in \mathsf{L}_s(X, \mathbb{Y})$, that is, $P_r(X) = \cdots = P_{s-1}(X) = 0$.

(b) $\deg(\widetilde{P}) \leq d + k - 1$, and $\deg(P_j) \leq d$ for all $j \in [0, r-1]$.

(c) $(\Delta^j P)\big(\alpha_i, w_{i,t}^{(0)}, \ldots, w_{i,t}^{(s-1)}\big) = 0$ for all $j \in [0, s-r]$, $t \in [\ell]$, $i \in [n]$.

The number of linear constraints is $n\ell(s-r+1)$, and the number of coefficients is $d + k + r(d+1) = d(r+1) + (r+k)$. So for the choice

$$d = \left\lceil \frac{n\ell(s-r+1) - (r+k) + 1}{r+1} \right\rceil,$$

we indeed have $d(r+1) + (r+k) > n\ell(s-r+1)$, and this ensures a nontrivial solution, that is, $P(X, \mathbb{Y}) \neq 0$.

Now consider any $f(X) \in \mathbb{K}[X]$, and suppose $\mathsf{d}_A^{(s)}(f, S) = 1 - (\nu/n)$. By Lemma 16, this implies $\sum_{a \in A} \mu_{\mathtt{Q}}(P^{[f]}, a) \geq \nu s - r + 1$. But we also have $\deg(P^{[f]}) \leq d + k - 1$. Therefore, by Proposition 12, we can conclude that $P^{[f]}(X) = 0$ if $\nu > \lceil (d+k-1)/(s-r+1) \rceil$. This is equivalent to

$$\nu > \left\lceil \frac{\left\lceil \frac{n\ell(s-r+1)-(r+k)+1}{r+1} \right\rceil + k - 1}{s-r+1} \right\rceil = \left\lceil \frac{n\ell(s-r+1) - (r+k) + (r+1)(k-1) + 1}{(r+1)(s-r+1)} \right\rceil.$$

The above is true if

$$\nu \geq \frac{n\ell}{r+1} + \frac{k}{s-r+1}, \quad \text{that is,} \quad \frac{\nu}{n} \geq \frac{\ell}{r+1} + \frac{k}{(s-r+1)n}.$$ ◀

Once we have an interpolating polynomial that captures the correct codewords, we can extract the output list as follows.

▶ **Proposition 18.** *Consider parameters $s \geq r \geq 1$, and $k, d \geq 1$ such that $d + k - 1 < [3]_q$. If $P(X, \mathbb{Y}) = \widetilde{P}(X) + \sum_{j=0}^{r-1} P_j(X) Y_j \in \mathsf{L}_s(X, \mathbb{Y})$ is a nonzero polynomial with*

$$\deg(\widetilde{P}) \leq d + k - 1, \quad \textit{and} \quad \deg(P_j) \leq d \qquad \textit{for all } j \in [0, r-1],$$

then the solution space

$$\{f(X) \in \mathbb{K}[X] : P^{[f]}(X) = 0\}$$

is an affine $\mathbb{K}$-linear space with dimension at most $r - 1$.

Proof. Consider any $f(X) \in \mathbb{K}[X]$ satisfying $P^{[f]}(X) = 0$. Note that we have $P(X, \mathbb{Y}) \neq 0$. If $P_j(X) = 0$ for all $j \in [0, r-1]$, then $P^{[f]}(X) = \widetilde{P}(X) = 0$, which means $P(X, \mathbb{Y}) = 0$, a contradiction. So there exists $j \in [0, r-1]$ such that $P_j(X) \neq 0$. Without loss of generality, we assume $P_{r-1}(X) \neq 0$. (Otherwise, we work with the largest r' such that $P_{r'-1}(X) \neq 0$.)

Let us also quickly consider another notation. For any $\mathtt{Q}^b \in \mathbb{K}^\times$, $h \geq 0$, and $g(X) \in \mathbb{K}[X]$, denote

$$g_{h,b} := \mathrm{coeff}\left(\frac{(X - \mathtt{Q}^b)_{\mathtt{Q}}^{(h)}}{[h]_{\mathtt{Q}}!}, g \right) = \mathsf{D}_{\mathtt{Q}}^h g(\mathtt{Q}^b) \qquad \text{by Proposition 6(b)}$$

Recall that $\mathbb{K} = \mathbb{F}_{q^3}$. Since $\deg(P_{r-1}) \leq d < [3]_q$, there exists $\mathtt{Q}^b \in \mathbb{K}^\times$ such that $P_{r-1}(\mathtt{Q}^{h+b}) \neq 0$ for all $h \in [0, d+k-1]$. Further, since $\deg(P) \leq d + k - 1 < [3]_q$, we will work with the basis of monomials $\{(X - \mathtt{Q}^b)_{\mathtt{Q}}^{(h)} : h \in [0, d+k-1]\}$. Now we have

$$P^{[f]}(X) \equiv \widetilde{P}(X) + P_0(X) f(X) + P_1(X) \mathsf{D}_{\mathtt{Q}} f(X) + \cdots + P_{r-1}(X) \mathsf{D}_{\mathtt{Q}}^{r-1} f(X) = 0$$

So by Proposition 6(c), for every $h \in [0, d+k-1]$, we have

$$\begin{aligned} 0 = P_{h,b}^{[f]} &= (\widetilde{P})_{h,b} + \sum_{j=0}^{r-1} \sum_{c=0}^{h} \begin{bmatrix} h \\ c \end{bmatrix}_{\mathtt{Q}} (P_j)_{h-c,c+b} (\mathsf{D}_{\mathtt{Q}}^j f)_{c,b} \\ &= (\widetilde{P})_{h,b} + \sum_{j=0}^{r-1} \sum_{c=0}^{h} \begin{bmatrix} h \\ c \end{bmatrix}_{\mathtt{Q}} (P_j)_{h-c,c+b} f_{c+j,b}. \end{aligned}$$

Now for every $h \in [0, d+k-1]$, since $(P_{r-1})_{0,h+b} = P_{r-1}(\mathtt{Q}^{h+b}) \neq 0$, we can rewrite the above

$$f_{h+r-1,b} = -\frac{1}{(P_{r-1})_{0,h+b}} \left((\widetilde{P})_{h,b} + \sum_{\substack{(j,c) \leq (r-1,h) \\ (j,c) \neq (r-1,h)}} \begin{bmatrix} h \\ c \end{bmatrix}_{\mathtt{Q}} (P_j)_{h-c,c+b} f_{c+j,b} \right).$$

So we can set the undetermined coefficients among $f_{0,b}, \ldots, f_{r-2,b} \in \mathbb{K}$ freely, and the remaining coefficients of $f(X)$ are uniquely determined the above equation. This means the solution space is an affine $\mathbb{K}$-linear space having dimension at most $r - 1$. ◀

The final list recovery result can now be proven.

Proof of Theorem 15. Choosing $r = \lceil \ell/\epsilon \rceil$, and plugging in all the parameters into Proposition 17 and Proposition 18 immediately implies that the output list is contained in an affine $\mathbb{K}$-linear space with dimension at most $\lceil \ell/\epsilon \rceil - 1$. The pruning algorithm with the improved analysis [44, 66] then asserts that the output list size is at most $(\ell/\epsilon)^{O((1+\log \ell)/\epsilon)}$. ◀

3.2 List decoding/recovery of multivariate $\mathtt{Q}$-multiplicity codes

We will now see an algorithm to list decode $\mathtt{Q}\text{-Mult}_{m,s}(A;k)$. Even though it is inspired by the classical multivariate multiplicity code list decoder [9], it turns out that we can give a simpler algorithm and analysis that is perhaps the correct multivariate extension of our list decoder in the univariate case. We will, in fact, perform list recovery, and the main theorem statement is the following.

▶ **Theorem 19.** *Consider any $\delta \in (0,1)$, $\epsilon \in (0, 1-\delta)$. For any $A \subseteq \mathbb{F}_q^\times$ and $\ell \geq 1$, and for the choices $s = O(\ell/\epsilon^{2m+1})$ and $k = \lceil (1-\delta)s|A| \rceil$, the code $\mathtt{Q}\text{-Mult}_s^m(A;k)$ is efficiently list recoverable up to radius $\delta - \epsilon$ with output list contained in a $\mathbb{K}$-affine space of dimension atmost $O(\ell^m/\epsilon^{2m})$.*

Note that setting $\ell = 1$ in Theorem 19 gives Theorem 2.

Consider additional indeterminates $\mathbb{Y} = (Y_\gamma)_{|\gamma|<s}$, $\mathbf{Z} = (Z_1, \ldots, Z_m)$, and the $\mathbb{K}$-linear subspace of the polynomial ring $\mathbb{K}[\mathbb{X}, \mathbb{Y}]$ defined by

$$\mathsf{L}_s(\mathbb{X}, \mathbb{Y}) = \left\{ \widetilde{P}(\mathbb{X}) + \sum_{j=0}^{s-1} P_j(\mathbb{X}) \left(\sum_{|\gamma|=j} Y_\gamma \mathbf{Z}^\gamma \right) \in \mathbb{K}(\mathbf{Z})[\mathbb{X}, \mathbb{Y}] \right\}.$$

Also, denote $Y_\gamma = 0$ for all $\gamma \in \mathbb{N}^m$, $|\gamma| \geq s$. For any $\alpha \in \mathbb{N}^m$, define a $\mathbb{K}$-linear operator $\Delta^{(\alpha)} : \mathsf{L}_s(\mathbb{X}, \mathbb{Y}) \to \mathsf{L}_s(\mathbb{X}, \mathbb{Y})$ by

$$\Delta^{(\alpha)} \left(\widetilde{P}(\mathbb{X}) + \sum_{j=0}^{s-1} P_j(\mathbb{X}) \left(\sum_{|\gamma|=j} Y_\gamma \mathbf{Z}^\gamma \right) \right) = \mathsf{D}_\mathtt{Q}^\alpha \widetilde{P}(\mathbb{X}) + \sum_{j=0}^{s-1} \sum_{|\gamma|=j} \left(\sum_{\beta \leq \alpha} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}_\mathtt{Q} \mathsf{D}_\mathtt{Q}^{\alpha-\beta} P_j(\mathtt{Q}^\beta \mathbb{X}) Y_{\beta+\gamma} \right) \mathbf{Z}^\gamma.$$

For any $P(\mathbb{X}, \mathbb{Y}, \mathbf{Z}) \in \mathsf{L}_s(\mathbb{X}, \mathbb{Y})$ and $f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]$, denote

$$P^{[f]}(\mathbb{X}) = P\left(\mathbb{X}, \left(\mathsf{D}_\mathtt{Q}^\gamma f(\mathbb{X}) \right)_{|\gamma|<s}, \mathbf{Z} \right) \in \mathbb{K}(\mathbf{Z})[\mathbb{X}].$$

The following is then immediate.

▶ **Lemma 20.** *Let $P(\mathbb{X}, \mathbb{Y}, \mathbf{Z}) \in \mathsf{L}_s(\mathbb{X}, \mathbb{Y})$.*

(a) *If $j \in [0, s-1]$ such that $\deg_{Y_\gamma}(P) = 0$ for all $\gamma \in \mathbb{N}^m$, $|\gamma| \geq j$, then $\deg_{Y_\gamma}(\Delta^{(\alpha)} P) = 0$ for all $\gamma \in \mathbb{N}^m$, $|\gamma| \geq j + |\alpha|$.*

(b) *$(\Delta^{(\alpha)} P)^{[f]}(\mathbb{X}) = \mathsf{D}_\mathtt{Q}^\alpha (P^{[f]})(\mathbb{X})$, for all $f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]$.*

▶ Remark 21. A consequence of Lemma 20 is that $\Delta^{(\alpha+\beta)} = \Delta^{(\alpha)} \circ \Delta^{(\beta)}$ for all $\alpha, \beta \in \mathbb{N}^m$. We do not explicitly use this elsewhere.

Proof of Lemma 20. Let $P(\mathbb{X}, \mathbb{Y}, \mathbf{Z}) = \widetilde{P}(\mathbb{X}) + \sum_{j=0}^{s-1} P_j(\mathbb{X}) \left(\sum_{|\gamma|=j} Y_\gamma \mathbf{Z}^\gamma \right) \in \mathsf{L}_s(\mathbb{X}, \mathbb{Y})$.

(a) If $j \in [0, s-1]$ such that $\deg_{Y_\gamma}(P) = 0$ for all $\gamma \in \mathbb{N}^m$, $|\gamma| \geq j$, this means $P_{j'}(\mathbb{X}) = 0$ for all $j' \geq j$. Now we have

$$\begin{aligned}
(\Delta^{(\alpha)} P)(\mathbb{X}, \mathbb{Y}, \mathbf{Z}) &= \mathsf{D}_\mathtt{Q}^\alpha \widetilde{P}(\mathbb{X}) + \sum_{j'=0}^{s-1} \sum_{|\gamma|=j'} \left(\sum_{\beta \leq \alpha} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}_\mathtt{Q} \mathsf{D}_\mathtt{Q}^{\alpha-\beta} P_{j'}(\mathtt{Q}^\beta \mathbb{X}) Y_{\beta+\gamma} \right) \mathbf{Z}^\gamma \\
&= \mathsf{D}_\mathtt{Q}^\alpha \widetilde{P}(\mathbb{X}) + \sum_{j'=0}^{s-1} \sum_{|\gamma|=j'} \left(\sum_{\beta \leq \alpha} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}_\mathtt{Q} \mathsf{D}_\mathtt{Q}^{\alpha-\beta} P_{j'-|\beta|}(\mathtt{Q}^\beta \mathbb{X}) \mathbf{Z}^{\gamma-\beta} \right) Y_\gamma
\end{aligned}$$

So, $\sum_{\beta \leq \alpha} \mathsf{D}_\mathtt{Q}^{\alpha-\beta} P_{j'-|\beta|}(\mathtt{Q}^\beta \mathbb{X}) \mathbf{Z}^{\gamma-\beta} = 0$ for all $j' \geq j + |\alpha|$. This means $\deg_{Y_{j'}}(\Delta^{(\alpha)} P) = 0$ for all $j' \geq j + |\alpha|$.

(b) We have

$$\begin{aligned}(\Delta^{(\alpha)}P)^{[f]}(\mathbb{X}) &= \mathsf{D}_{\mathtt{Q}}^{\alpha}\widetilde{P}(\mathbb{X}) + \sum_{j=0}^{s-1}\sum_{|\gamma|=j}\Big(\sum_{\beta\le\alpha}\begin{bmatrix}\alpha\\ \beta\end{bmatrix}_{\mathtt{Q}}\mathsf{D}_{\mathtt{Q}}^{\alpha-\beta}P_j(\mathtt{Q}^{\beta}\mathbb{X})\mathsf{D}_{\mathtt{Q}}^{\beta+\gamma}f(\mathbb{X})\Big)\mathbf{Z}^{\gamma}\\ &= \mathsf{D}_{\mathtt{Q}}^{\alpha}\widetilde{P}(\mathbb{X}) + \sum_{j=0}^{s-1}\sum_{|\gamma|=j}\mathsf{D}_{\mathtt{Q}}^{\alpha}\big(P_j\cdot\mathsf{D}_{\mathtt{Q}}^{\gamma}f\big)(\mathbb{X})\mathbf{Z}^{\gamma} \qquad \text{by Proposition 8(d)}\\ &= \mathsf{D}_{\mathtt{Q}}^{\alpha}(P^{[f]})(\mathbb{X}).\end{aligned}$$

◀

The interpolation step of the list recovery algorithm, which will be used multiple times in this discussion, can be captured as follows.

▶ **Proposition 22.** *Consider any $A\subseteq\mathbb{F}_q^{\times}$, parameters $s\ge r\ge 1$, and $k\in[s|A|]$. Let $\ell\ge 1$, and define*

$$d=\left\lceil 5\Big(\frac{\ell}{r+1}\Big)^{1/m}(s-r+1)|A|\right\rceil.$$

Then for any $S_a\subseteq\mathbb{K}^{\binom{m+s-1}{s-1}}$ with $|S_a|\le\ell$, $a\in A^m$, there exists a nonzero polynomial $P(\mathbb{X},\mathbb{Y},\mathbf{Z})=\widetilde{P}(\mathbb{X})+\sum_{j=0}^{r-1}P_j(\mathbb{X})\sum_{|\gamma|=j}Y_{\gamma}\mathbf{Z}^{\gamma}\in\mathsf{L}_s(\mathbb{X},\mathbb{Y})$ with

$$\deg(\widetilde{P})\le d+k-1,\quad \textit{and}\quad \deg(P_j)\le d\qquad \textit{for all } j\in[0,r-1].$$

such that for any $f(\mathbb{X})\in\mathbb{K}[\mathbb{X}]$, $\deg(f)<k$,

$$\textit{if}\quad \mathsf{d}_A^{(m,s)}(f,S)\ge 5\Big(\frac{\ell}{r+1}\Big)^{1/m}+\frac{k}{(s-r+1)|A|^{m-1}}.\qquad \textit{then}\quad P^{[f]}(\mathbb{X})=0.$$

Moreover, we can ensure that $P(\mathbb{X},\mathbb{Y},\mathbf{Z})$ is a polynomial in $\mathbf{Z}$ with $\deg_{\mathbf{Z}}(P)<ms$.

Proof. Assume the notation $u=(u^{(\gamma)}:|\gamma|<s)\in\mathbb{K}^{\binom{m+s-1}{s-1}}$. We can, in fact, add additional arbitrary elements and assume $|S_a|=\ell$, for all $a\in A^m$. Also enumerate $S_a=\{w_{a,1},\dots,w_{a,\ell}\}$, $a\in A^m$. We will construct a nonzero polynomial $P(\mathbb{X},\mathbb{Y},\mathbf{Z})=\widetilde{P}(\mathbb{X})+\sum_{j=0}^{s-1}P_j(\mathbb{X})\big(\sum_{|\gamma|=j}Y_{\gamma}\mathbf{Z}^{\gamma}\big)\in\mathsf{L}_s(\mathbb{X},\mathbb{Y})$ such that

(a) $P(\mathbb{X},\mathbb{Y},\mathbf{Z})=\widetilde{P}(\mathbb{X})+\sum_{j=0}^{r-1}P_j(\mathbb{X})\big(\sum_{|\gamma|=j}Y_{\gamma}\mathbf{Z}^{\gamma}\big)$, that is, $P_r(\mathbb{X})=\cdots=P_{s-1}(\mathbb{X})=0$.

(b) $\deg(\widetilde{P})\le d+k-1$, and $\deg(P_j)\le d$ for all $j\in[0,r-1]$.

(c) $(\Delta^{(\alpha)}P)\Big(a,\big(w_{a,t}^{(\gamma)}\big)_{|\gamma|<s}\Big)=0$ for all $\alpha\in\mathbb{N}^m$ with $|\alpha|\in[0,s-r]$, $t\in[\ell]$, $a\in A^m$.

The number of linear constraints is

$$|A|^m\ell\binom{m+s-r}{s-r}\le\ell\Big(\frac{5(s-r+1)|A|}{m}\Big)^m,$$

and the number of coefficients is

$$\binom{m+d+k-1}{d+k-1}+r\binom{m+d}{d}\ge(r+1)\binom{m+d}{d}\ge(r+1)\Big(\frac{d}{m}\Big)^m.$$

So for the choice

$$d=\left\lceil 5\Big(\frac{\ell}{r+1}\Big)^{1/m}(s-r+1)|A|\right\rceil,$$

we indeed get a nontrivial solution, that is, $P(\mathbb{X}, \mathbb{Y}, \mathbf{Z}) \neq 0$. Further, since each linear constraint is a polynomial in $\mathbf{Z} = (Z_1, \ldots, Z_m)$ having degree at most $s-1$, we can ensure that $P(\mathbb{X}, \mathbb{Y}, \mathbf{Z})$ is a polynomial in $\mathbf{Z}$ having $\deg_{\mathbf{Z}}(P) < ms$ (see [38] or [9, Lemma 3].)

Now consider any $f(X) \in \mathbb{K}[X]$, and suppose $\mathsf{d}_A^{(m,s)}(f, S) = 1-(\nu/|A|^{m-1})$. By Lemma 20, this implies $\sum_{a \in A^m} \mu_{\mathtt{Q}}(P^{[f]}, a) \geq \nu(s-r+1)$. But we also have $\deg(P^{[f]}) \leq (d+k-1)|A|^{m-1}$. Therefore, by Theorem 13, we can conclude that $P^{[f]}(\mathbb{X}) = 0$ if $\nu > (d+k-1)|A|^{m-1}/(s-r+1)$, which holds if

$$\nu > \left\lceil \frac{5(\ell/(r+1))^{1/m}(s-r+1)|A|^{m-1} + k - 1}{s-r+1} \right\rceil.$$

The above is true if

$$\nu \geq 5\left(\frac{\ell}{r+1}\right)^{1/m} |A|^{m-1} + \frac{k}{s-r+1},$$

$$\text{that is,} \quad \frac{\nu}{|A|^{m-1}} \geq 5\left(\frac{\ell}{r+1}\right)^{1/m} + \frac{k}{(s-r+1)|A|^{m-1}}. \qquad \blacktriangleleft$$

Once we have an interpolating polynomial that captures the correct codewords, we can extract the output list as follows.

▶ **Proposition 23.** *Consider parameters $s \geq r \geq 1$, and $k, d \geq 1$ such that $d + k - 1 < [3]_q$. If $P(\mathbb{X}, \mathbb{Y}, \mathbf{Z}) = \widetilde{P}(\mathbb{X}) + \sum_{j=0}^{r-1} P_j(\mathbb{X}) \sum_{|\gamma|=j} Y_\gamma \mathbf{Z}^\gamma \in \mathsf{L}_s(\mathbb{X}, \mathbb{Y}, \mathbf{Z})$ is a nonzero polynomial with*

$$\deg(\widetilde{P}) \leq d + k - 1, \quad \textit{and} \quad \deg(P_j) \leq d \qquad \textit{for all } j \in [0, r-1],$$

then the solution space

$$\{f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}] : P^{[f]}(\mathbb{X}) = 0\}$$

is an affine $\mathbb{K}$-linear space with dimension at most $\binom{m+r-2}{r-2}$.

Proof. Consider any $f(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]$ satisfying $P^{[f]}(\mathbb{X}) = 0$. Note that we have $P(\mathbb{X}, \mathbb{Y}, \mathbf{Z}) \neq 0$. If $P_j(\mathbb{X}) = 0$ for all $j \in [0, r-1]$, then $P^{[f]}(\mathbb{X}) = \widetilde{P}(\mathbb{X}) = 0$, which means $P(\mathbb{X}, \mathbb{Y}, \mathbf{Z}) = 0$, a contradiction. So there exists $j \in [0, r-1]$ such that $P_j(\mathbb{X}) \neq 0$. Without loss of generality, we assume $P_{r-1}(\mathbb{X}) \neq 0$. (Otherwise, we work with the largest r' such that $P_{r'-1}(\mathbb{X}) \neq 0$.)

Let us also quickly consider another notation. For any $\mathtt{Q}^\beta \in (\mathbb{K}^\times)^m$, $\alpha \in \mathbb{N}^m$, and $g(\mathbb{X}) \in \mathbb{K}[\mathbb{X}]$, denote

$$g_{\alpha,\beta} := \operatorname{coeff}\left(\frac{(\mathbb{X} - \mathtt{Q}^\beta)_{\mathtt{Q}}^{(\alpha)}}{[\alpha]_{\mathtt{Q}}!}, g\right) = \mathsf{D}_{\mathtt{Q}}^\alpha g(\mathtt{Q}^\beta) \qquad \text{by Proposition 8(b)}$$

Recall that $\mathbb{K} = \mathbb{F}_{q^3}$. Since $\deg(P_{r-1}) \leq d < [3]_q$, there exists $\mathtt{Q}^\beta \in (\mathbb{K}^\times)^m$ such that $P_{r-1}(\mathtt{Q}^{\alpha+\beta}) \neq 0$ for all $\alpha \in \mathbb{N}^m$, $|\alpha| \leq d+k-1$. Further, since $\deg(P) \leq d+k-1 < [3]_q$, we will work with the basis of monomials $\{(\mathbb{X} - \mathtt{Q}^\beta)_{\mathtt{Q}}^{(\alpha)} : \alpha \in \mathbb{N}^m, |\alpha| \leq d+k-1\}$. Now we have

$$P^{[f]}(\mathbb{X}) \equiv \widetilde{P}(\mathbb{X}) + \sum_{j=0}^{r-1} P_j(\mathbb{X}) \sum_{|\gamma|=j} \mathsf{D}_{\mathtt{Q}}^\gamma f(\mathbb{X}) \mathbf{Z}^\gamma = 0.$$

So for every $\alpha \in \mathbb{N}^m$, $|\alpha| \leq d+k-1$, we have

$$\begin{aligned} 0 = P_{\alpha,\beta}^{[f]} &= \widetilde{P}_{\alpha,\beta} + \sum_{j=0}^{r-1} \sum_{|\gamma|=j} \left(\sum_{\theta \leq \alpha} \begin{bmatrix} \alpha \\ \theta \end{bmatrix}_{\mathtt{Q}} (P_j)_{\alpha-\theta,\theta+\beta} \left(\mathsf{D}_{\mathtt{Q}}^\gamma f\right)_{\theta,\beta} \mathbf{Z}^\gamma \right) \\ &= \widetilde{P}_{\alpha,\beta} + \sum_{j=0}^{r-1} \sum_{\theta \leq \alpha} \begin{bmatrix} \alpha \\ \theta \end{bmatrix}_{\mathtt{Q}} (P_j)_{\alpha-\theta,\theta+\beta} \left(\sum_{|\gamma|=j} f_{\theta+\gamma,\beta} \mathbf{Z}^\gamma \right). \end{aligned}$$

Now for every $\alpha \in \mathbb{N}^m$, $|\alpha| \leq d+k-1$, since $P_{r-1}(\mathtt{Q}^{\alpha+\beta}) = (P_{r-1})_{0^m,\alpha+\beta} \neq 0$, we can rewrite the above as

$$\sum_{|\gamma|=r-1} f_{\alpha+\gamma,\beta}\mathbf{Z}^\gamma$$
$$= -\frac{1}{(P_{r-1})_{0^m,\alpha+\beta}}\left((\widetilde{P})_{\alpha,\beta} + \sum_{\substack{(j,\theta)\leq(r-1,\alpha)\\(j,\theta)\neq(r-1,\alpha)}} \begin{bmatrix}\alpha\\ \theta\end{bmatrix}_{\mathtt{Q}} (P_j)_{\alpha-\theta,\theta+\beta}\left(\sum_{|\gamma|=j} f_{\theta+\gamma,\beta}\mathbf{Z}^\gamma\right)\right) \tag{3}$$

So we can set the undetermined coefficients among $f_{\gamma,\beta} \in \mathbb{K}$, $|\gamma| \leq r-2$ freely, and the remaining coefficients of $f(\mathbb{X})$ are uniquely determined the above equation. This means the solution space is an affine $\mathbb{K}$-linear space having dimension at most $\binom{m+r-2}{r-2}$. Further, at any instance of (3), if we know the R.H.S. and determine the L.H.S., then each coefficient $f_{\alpha+\gamma,\beta}$ can be recovered by instantiating an arbitrary finite grid S^m with $S \subseteq \mathbb{K}$, $|S| \geq ms$ as a hitting set for the $\mathbf{Z}$-variables, since we have $\deg_{\mathbf{Z}}(P) < ms$. ◀

We are now ready to complete the proof of our main theorem.

Proof of Theorem 19. Choosing $r = O(\ell/\epsilon^m)$, and plugging in all the parameters into Proposition 17 and Proposition 18 immediately proves the claim. ◀

References

1 Noga Alon. Combinatorial Nullstellensatz. *Combinatorics, Probability and Computing*, 8(1-2):7–29, 1999. doi:10.1017/S0963548398003411.

2 Omar Alrabiah, Venkatesan Guruswami, and Ray Li. Randomly Punctured Reed–Solomon Codes Achieve List-Decoding Capacity over Linear-Sized Fields. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, pages 1458–1469, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3618260.3649634.

3 G.E. Andrews. *q-Series: Their Development and Application in Analysis, Number Theory, Combinatorics, Physics and Computer Algebra*, volume 66 of *CBMS Regional Conference Lecture Series in Mathematics*. AMS, 1986.

4 M.H. Annaby and Z.S. Mansour. q-Taylor and interpolation series for Jackson q-difference operators. *Journal of Mathematical Analysis and Applications*, 344(1):472–483, 2008. doi:10.1016/j.jmaa.2008.02.033.

5 László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*, STOC '91, pages 21–32, New York, NY, USA, 1991. Association for Computing Machinery. doi:10.1145/103418.103428.

6 Simeon Ball and Oriol Serra. Punctured combinatorial nullstellensätze. *Combinatorica*, 29(5):511–522, 2009. doi:10.1007/s00493-009-2509-z.

7 Eli Ben-Sasson, Swastik Kopparty, and Jaikumar Radhakrishnan. Subspace Polynomials and Limits to List Decoding of Reed–Solomon Codes. *IEEE Transactions on Information Theory*, 56(1):113–120, 2010. doi:10.1109/TIT.2009.2034780.

8 Siddharth Bhandari, Prahladh Harsha, Mrinal Kumar, and Ashutosh Shankar. Algorithmizing the Multiplicity Schwartz-Zippel Lemma. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 2816–2835. SIAM, 2023. doi:10.1137/1.9781611977554.ch106.

9 Siddharth Bhandari, Prahladh Harsha, Mrinal Kumar, and Madhu Sudan. Decoding Multivariate Multiplicity Codes on Product Sets. *IEEE Transactions on Information Theory*, 70(1):154–169, 2024. doi:10.1109/TIT.2023.3306849.

10 George Boole. *A Treatise on the Calculus of Finite Differences*. MacMillan and Co., 1860. Reprint: Cambridge University Press, 2009. doi:10.1017/CBO9780511693014.

11 Joshua Brakensiek, Sivakanth Gopi, and Visu Makam. Generic Reed-Solomon Codes Achieve List-Decoding Capacity. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pages 1488–1501, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3564246.3585128.

12 Yeyuan Chen and Zihan Zhang. Explicit Folded Reed-Solomon and Multiplicity Codes Achieve Relaxed Generalized Singleton Bound. *arXiv Preprint*, 2024. doi:10.48550/arXiv.2408.15925.

13 David Cox, John Little, and Donald O'Shea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013.

14 P. Delsarte, J.M. Goethals, and F.J. Mac Williams. On generalized Reed Muller codes and their relatives. *Information and Control*, 16(5):403–442, 1970. doi:10.1016/S0019-9958(70)90214-7.

15 Zeev Dvir, Swastik Kopparty, Shubhangi Saraf, and Madhu Sudan. Extensions to the Method of Multiplicities, with Applications to Kakeya Sets and Mergers. *SIAM Journal on Computing*, 42(6):2305–2328, 2013. doi:10.1137/100783704.

16 Peter Elias. *List Decoding for Noisy Channels*. Ph.d. thesis, Massachusetts Institute of Technology, 1957. RLE Technical Report No. 335. http://hdl.handle.net/1721.1/4484.

17 Thomas Ernst. *A Comprehensive Treatment of q-Calculus*. Birkhäuser Basel, 2012. doi:10.1007/978-3-0348-0431-8.

18 H. Exton. *q-Hypergeometric Functions and Applications*. Halsted Press, Chichister, 1983.

19 Bálint Felszeghy, Balázs Ráth, and Lajos Rónyai. The lex game and some applications. *Journal of Symbolic Computation*, 41(6):663–681, 2006. doi:10.1016/j.jsc.2005.11.003.

20 Dominique Foata and Guo-Niu Han. The q-Series in Combinatorics: Permutation Statistics (Lecture addendums, revised version). *Unpublished*, 2021. Available at https://irma.math.unistra.fr/~foata/paper/pub91.pdf.

21 George Gasper and Mizan Rahman. *Basic Hypergeometric Series*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2 edition, 2004. doi:10.1017/CBO9780511526251.

22 Carl Friedrich Gauß. *Summatio quarumdam serierum singularium*. Dieterich, 1808. Digitized version available at http://eudml.org/doc/203313.

23 O. Geil and U. Martínez-Peñas. Bounding the Number of Common Zeros of Multivariate Polynomials and their Consecutive Derivatives. *Combinatorics, Probability and Computing*, 28(2):253–279, 2019. doi:10.1017/S0963548318000342.

24 Rohan Goyal, Prahladh Harsha, Mrinal Kumar, and Ashutosh Shankar. Fast list-decoding of univariate multiplicity and folded Reed-Solomon codes. *arXiv Preprint*, 2023. doi:10.48550/arXiv.2311.17841.

25 Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. Derandomization from algebraic hardness. *SIAM Journal on Computing*, 51(2):315–335, 2022. doi:10.1137/20M1347395.

26 Zeyu Guo and Zihan Zhang. Randomly Punctured Reed-Solomon Codes Achieve the List Decoding Capacity over Polynomial-Size Alphabets. *arXiv Preprint*, 2023. doi:10.48550/arXiv.2304.01403.

27 V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280)*, pages 28–37, 1998. doi:10.1109/SFCS.1998.743426.

28 Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Transactions on Information Theory*, 54(1):135–150, 2008. doi:10.1109/TIT.2007.911222.

29 Venkatesan Guruswami and Carol Wang. Linear-algebraic list decoding for variants of reed–solomon codes. *IEEE Transactions on Information Theory*, 59(6):3257–3268, 2013. doi:10.1109/TIT.2013.2246813.

30 Wolfgang Hahn. Über Orthogonalpolynome, die q-Differenzengleichungen genügen. *Mathematische Nachrichten*, 2(1-2):4–34, 1949. doi:10.1002/mana.19490020103.

31 F. H. Jackson. A q-form of Taylor's theorem. *Messenger of Mathematics*, 38:62–64, 1909.

32 F. H. Jackson. A q-series corresponding to Taylor's series. *Messenger of Mathematics*, 39:26–28, 1909.

33 F. H. Jackson. On q-Functions and a certain Difference Operator. *Transactions of the Royal Society of Edinburgh*, 46(2):253–281, 1909. doi:10.1017/S0080456800002751.

34 F. H. Jackson. On q-Definite Integrals. *The Quarterly Journal of Pure and Applied Mathematics*, 41:193–203, 1910.

35 F. H. Jackson. q-Difference Equations. *American Journal of Mathematics*, 32(4):305–314, 1910. doi:10.2307/2370183.

36 Charles Jordan. *Calculus of Finite Differences*. Chelsea Publishing Company, New York, 2nd edition, 1950.

37 Victor Kac and Pokman Cheung. *Quantum Calculus*. Springer New York, NY, 2002. doi:10.1007/978-1-4613-0071-7.

38 R. Kannan. Solving systems of linear equations over polynomials. *Theoretical Computer Science*, 39:69–88, 1985. Third Conference on Foundations of Software Technology and Theoretical Computer Science. doi:10.1016/0304-3975(85)90131-8.

39 T. Kasami, Shu Lin, and W. Peterson. New generalizations of the Reed-Muller codes–I: Primitive codes. *IEEE Transactions on Information Theory*, 14(2):189–199, 1968. doi:10.1109/TIT.1968.1054127.

40 T. Kasami, Shu Lin, and W. Peterson. Polynomial codes. *IEEE Transactions on Information Theory*, 14(6):807–814, 1968. doi:10.1109/TIT.1968.1054226.

41 John Y. Kim and Swastik Kopparty. Decoding Reed-Muller Codes over Product Sets. *Theory of Computing*, 13(21):1–38, 2017. doi:10.4086/toc.2017.v013a021.

42 Wolfram Koepf, Predrag M. Rajković, and Sladjana D. Marinković. Properties of q-holonomic functions. *Journal of Difference Equations and Applications*, 13(7):621–638, 2007. doi:10.1080/10236190701264925.

43 Swastik Kopparty. List-Decoding Multiplicity Codes. *Theory of Computing*, 11(5):149–182, 2015. URL: 10.4086/toc.2015.v011a005, doi:10.4086/TOC.2015.V011A005.

44 Swastik Kopparty, Noga Ron-Zewi, Shubhangi Saraf, and Mary Wootters. Improved List Decoding of Folded Reed-Solomon and Multiplicity Codes. *SIAM Journal on Computing*, 52(3):794–840, 2023. doi:10.1137/20M1370215.

45 Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-Rate Codes with Sublinear-Time Decoding. *J. ACM*, 61(5), 2014. doi:10.1145/2629416.

46 S.F. Lacroix. *Traité des Différences et des Séries*. Vol. 3 of *Traité du Calcul Differentiel et du Calcul Intégral*. Courcier, Paris, 1819.

47 D. Larsson and S.D. Silvestrov. Burchnall-Chaundy Theory for q-Difference Operators and q-Deformed Heisenberg Algebras. *Journal of Nonlinear Mathematical Physics*, 10(sup2):95–106, 2003. doi:10.2991/jnmp.2003.10.s2.7.

48 Rudolf Lidl and Harald Niederreiter. *Finite Fields*. Cambridge University Press, 2nd edition, 1997.

49 J. Massey. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, 15(1):122–127, 1969. doi:10.1109/TIT.1969.1054260.

50 Tamás Mészáros. *S-extremal set systems and Gröbner bases* (Diploma Thesis). PhD thesis, Budapest University of Technology and Economics, 2005. Available at https://sites.google.com/view/tamas-meszaros/.

51 Shay Moran and Cyrus Rashtchian. Shattered Sets and the Hilbert Function. In Piotr Faliszewski, Anca Muscholl, and Rolf Niedermeier, editors, *41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016)*, volume 58 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 70:1–70:14, Dagstuhl, Germany, 2016. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2016.70.

52 D.E. Muller. Application of Boolean algebra to switching circuit design and to error detection. *Transactions of the I.R.E. Professional Group on Electronic Computers*, EC-3(3):6–12, 1954. doi:10.1109/IREPGELC.1954.6499441.

53 Rasmus Refslund Nielsen. *List decoding of linear block codes*. Ph.d. thesis, Technical University of Denmark, 2001. Available at https://orbit.dtu.dk/files/3028444/List%20decoding%20of%20linear%20block%20codes.pdf.

54 Niels Erik Nörlund. *Vorlesungen über Differenzenrechnung*. Springer Berlin, 1924.

55 R. Pellikaan and Xin-Wen Wu. List decoding of q-ary Reed-Muller codes. *IEEE Transactions on Information Theory*, 50(4):679–682, 2004. doi:10.1109/TIT.2004.825043.

56 W. Peterson. Encoding and error-correction procedures for the Bose-Chaudhuri codes. *IRE Transactions on Information Theory*, 6(4):459–470, 1960. doi:10.1109/TIT.1960.1057586.

57 I. Reed. A class of multiple-error-correcting codes and the decoding scheme. *Transactions of the IRE Professional Group on Information Theory*, 4(4):38–49, 1954. doi:10.1109/TIT.1954.1057465.

58 I. S. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960. doi:10.1137/0108018.

59 C.H. Richardson. *An Introduction to the Calculus of Finite Differences*. D. Van Nostrand Company, Inc., New York, 1954.

60 Steven Roman. *The Umbral Calculus*. Dover Publications, 2005.

61 Noga Ron-Zewi, S. Venkitesh, and Mary Wootters. Efficient List Decoding of Polynomial Ideal Codes with Optimal List Size. *arXiv Preprint*, 2024. doi:10.48550/arXiv.2401.14517.

62 Eugene Spiegel and Christopher O'Donnell. *Incidence Algebras*. CRC Press, 1997.

63 Shashank Srivastava. Improved List Size for Folded Reed-Solomon Codes. *arXiv Preprint*, 2024. doi:10.48550/arXiv.2410.09031.

64 J.F. Steffensen. *Interpolation*. The Williams & Wilkins Company, Baltimore, 1927.

65 Madhu Sudan. Decoding of Reed Solomon Codes beyond the Error-Correction Bound. *Journal of Complexity*, 13(1):180–193, 1997. doi:10.1006/jcom.1997.0439.

66 Itzhak Tamo. Tighter list-size bounds for list-decoding and recovery of folded Reed-Solomon and multiplicity codes. *IEEE Transactions on Information Theory (Early Access)*, pages 1–1, 2024. doi:10.1109/TIT.2024.3402171.

67 L. Welch. Error Correction for Algebraic Block Codes. *IEEE Int. Symp. Inform. Theory*, 1983. URL: https://cir.nii.ac.jp/crid/1573668924281711104.

68 E. Weldon. New generalizations of the Reed-Muller codes–II: Nonprimitive codes. *IEEE Transactions on Information Theory*, 14(2):199–205, 1968. doi:10.1109/TIT.1968.1054128.

69 J.M. Wozencraft. List Decoding. *Quarterly Progress Report, Research Lab of Electronics, MIT*, 48:90–95, 1958.

New Direct Sum Tests

Alek Westover
MIT, Cambridge, MA, USA

Edward Yu
MIT, Cambridge, MA, USA

Kai Zhe Zheng
MIT, Cambridge, MA, USA

Abstract

A function $f : [n]^d \to \mathbb{F}_2$ is a **direct sum** if there are functions $L_i : [n] \to \mathbb{F}_2$ such that $f(x) = \sum_i L_i(x_i)$. In this work we give multiple results related to the property testing of direct sums.

Our first result concerns a test proposed by Dinur and Golubev in [13]. We call their test the Diamond test and show that it is indeed a direct sum tester. More specifically, we show that if a function f is ε-far from being a direct sum function, then the Diamond test rejects f with probability at least $\Omega_{n,\varepsilon}(1)$. Even in the case of $n = 2$, the Diamond test is, to the best of our knowledge, novel and yields a new tester for the classic property of affinity.

Apart from the Diamond test, we also analyze a broad family of direct sum tests, which at a high level, run an arbitrary affinity test on the restriction of f to a random hypercube inside of $[n]^d$. This family of tests includes the direct sum test analyzed in [13], but does not include the Diamond test. As an application of our result, we obtain a direct sum test which works in the online adversary model of [20].

Finally, we also discuss a Fourier analytic interpretation of the diamond tester in the $n = 2$ case, as well as prove local correction results for direct sum as conjectured by [13].

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases Linearity testing, Direct sum, Grids

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.94

Related Version *arXiv Version*: https://arxiv.org/abs/2408.16092

Funding *Alek Westover*: Supported by MIT SPUR.
Edward Yu: Supported by MIT SPUR.
Kai Zhe Zheng: Department of Mathematics, Massachusetts Institute of Technology, Cambridge, USA. Supported by the NSF GRFP DGE-2141064.

1 Introduction

In property testing, one is given query access to a function over a large, typically multidimensional domain, say $f : \prod_{i=1}^{d} S_i \to T$. The goal is to determine whether or not f satisfies some property, $\mathcal{P}$, using as few queries to f as possible. Here, a property $\mathcal{P}$ can be any subset of functions. It is not hard to see that distinguishing between $f \in \mathcal{P}$ and $f \notin \mathcal{P}$ requires querying the entire domain for any nontrivial property $\mathcal{P}$, and so we typically settle for an algorithm with a weaker guarantee, typically called a property tester. A tester for $\mathcal{P}$ is a randomized algorithm, which given input function f, makes queries to f and satisfies the following two properties:

- **Completeness**: If $f \in \mathcal{P}$, the tester accepts with probability 1.
- **Soundness**: If f is δ-far from $\mathcal{P}$, the tester rejects with probability $s(\delta)$.

© Alek Westover, Edward Yu, and Kai Zhe Zheng;
licensed under Creative Commons License CC-BY 4.0

We say that f is δ-far from $\mathcal{P}$, if the minimal fractional hamming distance between f and any $g \in \mathcal{P}$ is at least δ. The function $s(\delta)$ is referred to as the soundness of the tester and oftentimes one repeats the tester in $s(\delta)^{-1}$ times to reject with probability $2/3$ in the soundness case. It is clear that for any property, there is a trivial algorithm that queries f on its entire domain, so oftentimes the goal is to design testers whose query complexity is sublinear in the domain size or even independent of the dimension d.

In this paper we consider two property testing questions related to functions known as direct sums. A function f is called a direct sum if it can be written as a sum of functions on the individual coordinates, that is

$$f(x_1, \ldots, x_d) = \sum_{i=1}^{d} L_i(x_i),$$

for some functions $L_i : S_i \to T$. Direct sums are a natural property to consider in the context of property testing. Indeed, when one takes $f : \mathbb{F}_2^d \to \mathbb{F}_2$, then the direct sum property is equivalent to the classic property of affinity (being a linear function plus a constant) considered by Blum, Luby, and Rubinfeld's seminal work [8]. Property testing of more general versions of direct sums, called low junta-degree functions, has also been considered before in the works of [5, 3]. A junta-degree t function is one that can be written as the sum of t-juntas. Thus, direct sums are junta-degree 1 functions, and direct sum testing is a specialization of the low junta-degree testing problem that appears in [5, 3].

In addition to being a natural property to study, motivation for direct sum testing also comes from its relation to the more widely known direct product testing problem. Historically, direct product testing was first introduced by Goldreich and Safra in [17] due to its potential for application as a hardness amplification step in more efficient PCP constructions [15, 12, 14, 18, 4]. More recently, the related problem of direct sum testing was considered, first by [10] and later by [13]. The motivation for considering direct sums stems in part from the fact that direct sums are similar to direct products in that they can be used to amplify hardness, but with the advantage that the output is a single value, as opposed to an entire tuple. This seems to make direct sum testing more difficult, but has the potential for leading to even more efficient PCPs (in particular, in improving a parameter known as the alphabet size). Analysis of a type of direct sum test is a key piece of Chan's elegant PCP construction in [9].

Direct Sum Testing. In [13], Dinur and Golubev give and analyze a direct sum tester which they call the Square in a Cube test. They propose an additional test which they refer to as the diamond test, but leave its analysis for future work. We call this second test the Diamond Test, and describe both of these tests below.

Let the domain be an arbitrary grid $[\overline{n}; d] = \prod_{i=1}^{d} [n_i]$ where each $[n_i] = \{1, \ldots, n_i\}$. We will let the output be $\mathbb{F}_2$ so that our input function is

$$f : [\overline{n}; d] \to \mathbb{F}_2.$$

Given two points $a, b \in [\overline{n}; d]$ and $x \in \{0,1\}^d$, let **interpolation** $\phi_x(a, b)$ to be the vector obtained by taking a_i whenever $x_i = 0$ and b_i whenever $x_i = 1$. The Square in a Cube test proceeds as follows.

▶ **Test 1** (Square in a Cube)**.** Sample random $a, b \in [\overline{n}; d]$ and $x, y \in \{0,1\}^d$. Accept if and only if

$$f(a) + f(\phi_x(a, b)) + f(\phi_y(a, b)) + f(\phi_{x+y}(a, b)) = 0.$$

The Diamond Test proceeds in a slightly different manner, which essentially fixes $x + y = (1, \ldots, 1)$.

▶ **Test 2** (Diamond). Sample random $a, b \in [\overline{n}; d], x \in \{0,1\}^d$. Accept iff

$$f(a) + f(\phi_x(a,b)) + f(\phi_x(b,a)) + f(b) = 0.$$

In addition to the Diamond test, we also consider a family of tests that generalizes the Square in a Cube test. We call this test the Affine on Subcube test.

▶ **Test 3** (Affine on Subcube). Sample $a, b \in [n]^d$ and run an affinity test on the function $x \mapsto f(\phi_x(a,b))$.

Note that the affinity test can be any arbitrary affinity test for functions $\mathbb{F}_2^d \to \mathbb{F}_2$. The Square in a Cube test is a special case of the above with the classic BLR affinity test as the affinity test. The Diamond Test, however, is not a specialization of the above.

As is usual in property testing, it is clear that both the Diamond and Affine on Subcube test satisfy completeness, so our main interest is with regards to their soundness.

1.1 Main Results

We now describe our main results. Our first result establishes soundness for the Diamond test. As the completeness is clear, this shows that the Diamond test is indeed a Direct Sum tester.

▶ **Theorem 4.** *Suppose $f : [n]^d \to \mathbb{F}_2$ passes the Diamond test with probability $1 - \varepsilon$. Then, f is $C_n \cdot \varepsilon$-close to a direct sum, for some constant C_n independent of d.*

We remark that the theorem above incurs a dependence on n. In contrast, [13] shows a version of the above result with C_n replaced by an absolute constant, C, independent of n. For $n = O(1)$, the soundness we show for the Diamond Test is comparable to what is known for the Square in a Cube test, but for larger n the soundness analysis becomes weaker. This dependence on n in the soundness is not so uncommon in property testing results however. It is comparable to the dependence on field size in the low degree testing results of [2, 21], or on grid size in the junta degree testing results of [5, 3]. We leave it to future work to remove the dependence on n in the soundness.

Our next result is a reformulation of Theorem 4 in the $n = 2$ case. Here the Diamond Test is, to the best of our knowledge, a novel affinity tester. Moreover, we observe that the soundness of the $n = 2$ Diamond Test is equivalent to a succinct Fourier analytic fact regarding a function's distance to affinity, or equivalently its maximum magnitude Fourier coefficient. Interestingly, we were unable to show this fact without appealing to the soundness of the Diamond test. Intuitively, the Fourier analytic fact says "f is close to affine if and only if when you randomly fix or don't fix each coordinate of f to a random value, f becomes close to an even or odd function". Formally, the result is:

▶ **Theorem 5.** *For $f : \mathbb{F}_2^d \to \mathbb{F}_2$, define $\mathcal{R}(f)$ to be a random restriction of f obtained as follows: for each $i \in [d]$ independently, with probability $1/4$ restrict x_i to 0, with probability $1/4$ restrict x_i to 1, and with probability $1/2$ do not restrict x_i. Then, letting* EvenOrOdd *denote the set of functions that are either even or odd, we have:*

$$\mathsf{dist}(f, \mathsf{affine}) = \Theta(\mathbb{E}_{\mathcal{R}}[\mathsf{dist}(\mathcal{R}(f), \mathsf{EvenOrOdd})]).$$

Along with the Diamond Test, we also describe and analyze a family of testers for direct sum which we call the Affine on Subcube test (Test 3).

▶ **Theorem 6.** *Fix $f : [n]^d \to \mathbb{F}_2, \varepsilon \geq 0$ and an affinity test T. If f passes the Affine on Subcube test (instantiated with T) with probability $1 - \varepsilon$, then f is $C\varepsilon$-close to a direct sum, for some constant C independent of n, d.*

By using a suitable erasure-resilient tester as the inner affinity tester in Test 3, we use Theorem 36 obtain direct sum testers in the online-erasure model recently introduced by [20]. We discuss this result as well as the online-erasure model in more detail in Section 4.2.

Finally, we address a second question posed in [13] regarding reconstructing functions which are close to direct sums. They ask if a test that they call the Shapka test can be used to reconstruct, via a "voting scheme", a direct sum given query access to its corrupted version. We show that this is indeed the case, and also give an improved reconstruction method that uses fewer queries.

▶ **Proposition 7.** *Let $f : [n]^d \to \mathbb{F}_2$ be ε close to a direct sum L, with $(n+1)\varepsilon < 1/4$. Then L can be reconstructed using a voting scheme on f using n queries to f.*

We also give lower bounds on the query complexity needed and show that our improved method is asymptotically optimal (up to constant factors) in terms of both query complexity and the fraction of corruptions it can tolerate.

Proof Overview. We give an overview of the proof of Theorem 4. Our analysis here diverges significantly from that of Dinur and Golubev for the Square in a Cube Test. Before discussing our analysis, we first summarize their approach and where it fails for the Diamond Test.

The underlying idea behind the Square in a Cube Test is that after choosing two points $a, b \in [n]^d$, any direct sum restricted to the hypercube-like domain $\prod_{i=1}^{d}\{a_i, b_i\}$ must be an affine function (i.e multivariate polynomial of degree 1). This idea gives way to a natural interpretation of the Square in a Cube Test

- Choose two points $a, b \in [n]^d$ randomly and view the domain $\prod_{i=1}^{d}\{a_i, b_i\}$ as a hypercube of the appropriate dimension d'. Here d' is the number of coordinates on which a and b differ.
- Define the function $f_{a,b} : \{0,1\}^{d'} \to \mathbb{F}_2$ by $f_{a,b}(x) = f(\phi_x(a, b))$.
- Perform the BLR affinity test on $f_{a,b}$. That is, choose $x, y, z \in \{0,1\}^{d'}$ and accept if and only if

$$f_{a,b}(x) + f_{a,b}(x+y) + f_{a,b}(x+z) + f_{a,b}(x+y+z) = 0.$$

From here, the analysis of [13] has two parts. First, they apply BLR to show that if the overall Square in a Cube test passes with high probability, then for many a, b, the function $f_{a,b}$ is close to an affine function (or direct sum), say $F_{a,b}$, on the hypercube $\prod_{i=1}^{d}\{a_i, b_i\}$. These $F_{a,b}$ can be thought of as local direct sums that agree with f locally on hypercubes inside of $[n]^d$. The second step is to show that many of these $F_{a,b}$ are actually consistent with each other and apply a direct product testing result of [16] to conclude that there is in fact a global direct sum F consistent with many $F_{a,b}$'s. As the $F_{a,b}$'s are in turn largely consistent with f, we get that F is a direct sum close to f, concluding the proof of soundness.

Our analysis of the soundness of the Diamond test, however, requires a different approach. Let us first see what happens when we try to naively adapt the above strategy. After choosing $a, b \in [n]^d$, instead of applying BLR, the Diamond test performs the following test on $f_{a,b} : \{0,1\}^{d'} \to \mathbb{F}_2$.

Choose $x \in \{0,1\}^{d'}$ and accept if and only if $f_{a,b}(0) + f_{a,b}(x) + f_{a,b}(x + \mathbb{1}) + f_{a,b}(\mathbb{1}) = 0$.

There is an issue though: the above is not an affinity tester! Indeed one can check that if $f_{a,b}(0) + f_{a,b}(\mathbb{1}) = 0$, resp. 1, then any even, resp. odd, function passes the above test with probability 1. Thus, we would only get that many $f_{a,b}$ are close to even or odd functions, and from here it is not clear how to make the second part of the analysis go through.

We instead combine ideas from the soundness analyses of [10, 7, 3], which use induction. Suppose $f : [n]^d \to \mathbb{F}_2$ is ε-far from a direct sum. We wish to show that the Diamond Test rejects f with some non-trivial probability that is independent of d. The inductive approach of both of the mentioned works consist of three main steps: 1) Restate the test on f as choosing a random subdomain that is one dimension and performing the test on f restricted to this subdomain, 2) analyze the soundness in the ε very small case, 3) on the other hand, show that if ε is large then show that f must also be far from a direct sum on many subdomains.

To execute all of the above steps, we have to draw on ideas from all of the mentioned works. For step 1), we take inspiration from [10] and instead analyze a four-function version of the Diamond Test. The reason is that there is no clear way to view the Diamond Test in the manner described above. It is tempting to describe the Diamond Test as choosing a random coordinate, say $i = 1$, and a random value in $[n]$, say 1, and then performing a $d-1$-dimensional version of the Diamond Test on the function $f(1, x_2, \ldots, x_d)$. Unfortunately this does not work as one has to change the distribution that the points in the $d-1$-dimensional grid are chosen. Indeed, if one chooses the points $a, b \in [n]^{d-1}$ of the one dimension down Diamond Test uniformly at random, then the final d-dimensional points output would agree on $1 + (d-1)/n$ coordinates in expectation instead of d/n.

For the small distance case in step 2), we rely on the a hypercontractivity theorem over grids. However, since we are now working with a four-function version of the Diamond Test some additional care is needed in this analysis.

Finally for step 3), we use techniques from agreement testing in the PCP literature [25, 23]. From [25] we borrow their transitive-consistency graph technique used to analyze the so-called Plane versus Plane test and from [23] we show a version of their hyperplane sampling lemma for $(d-1)$-dimensional grids in $[n]^d$.

Paper Outline. In Section 3 we present and analyze our novel affinity tester (the Diamond test). In Section 4 we give a new class of direct sum tests. In Section 5 we consider the problem of obtaining "corrected" samples from corrupted direct sums.

2 Preliminaries

We now introduce some notations that will be used throughout the paper. We let $[n]$ denote $\{1, 2, \ldots, n\}$. Let $[n; d]$ denote a product space $[n_1] \times \cdots \times [n_d]$. For a set or event X, we write $\overline{X}$ to denote the complement of X.

The notion of distance that we use is fractional hamming distance. That is, given two functions $f : S \to T$ and $g : S \to T$, the distance between them is the fraction of entries on which they differ:

$$\mathsf{dist}(f, g) = \Pr_{x \in S}[f(x) \neq g(x)].$$

Given a property $\mathcal{P} \subseteq \{g : S \to T\}$, the distance from f to the property $\mathcal{P}$ is defined as

$$\mathsf{dist}(f, \mathcal{P}) = \min_{g \in \mathcal{P}} \mathsf{dist}(f, g).$$

We say that f is δ-*far* from $\mathcal{P}$ if $\mathsf{dist}(f, \mathcal{P}) \geq \delta$.

Finally, if we do not explicitly specify the distribution a random variable is drawn from is a probability, then the random variable is meant to be drawn uniformly from the appropriate space (which will be clear from context). The constants hidden in our O-notation are never allowed to depend on d. If there is a parameter k that we think of as constant, we will sometimes use the notation $O_k(\cdot)$ to emphasize that the constant hidden by the O is allowed to depend on k.

Boolean Functions. Define the **interpolation** $\phi_x(a,b)$ to be the vector obtained by taking a_i whenever $x_i = 0$ and b_i whenever $x_i = 1$. A frequently useful property of interpolations is:

▶ **Fact 8.** $\phi_x(a,b) = \phi_{x+\mathbb{1}}(b,a)$.

Given $x \in \prod_i S_i$ and $p \in [0,1]$, let $y \sim \mathsf{T}_p(x)$ denote a vector in $\prod_i S_i$, with each y_i sampled independently as follows: set $y_i = x_i$ with probability p, and otherwise sample y_i uniformly from S_i; T_p is called the **noise operator**. Given $p \in [-1,0]$ and $x \in \mathbb{F}_2^d$, we define the distribution $y \sim \mathsf{T}_p(x)$ as: $y_i = x_i + 1$ with probability $-p$ and y_i is sampled uniformly from $\{0,1\}$ otherwise. For a boolean function $f : \mathbb{F}_2^d \to \mathbb{F}_2$ we will use $\mu(f)$ to denote the fractional size of the support of f (i.e., the fraction of $\mathbb{F}_2^d$ on which f outputs 1). We use $\mathbb{1}$ to denote the all ones vector, and sometimes use 0 to denote the zero-vector (the dimension will be clear from context). For $v, w \in \mathbb{F}_2^d$ we write $\langle v, w\rangle$ to denote $\sum_i v_i w_i$. For boolean functions $f, g : \mathbb{F}_2^d \to \mathbb{F}_2$ we write $\langle f, g\rangle$ to denote $\mathbb{E}_x[f(x)g(x)]$.

When discussing functions $f : [n]^d \to \mathbb{F}_2$ we will always assume $n > 1$.

3 The Diamond Test

We restate the Diamond test, initially proposed by Dinur and Golubev in [13] (recall from Section 2 that ϕ_x is the interpolation function)

▶ **Test 9** (Diamond)**.** Sample random $a, b \sim [n]^d, x \sim \mathbb{F}_2^d$. Accept iff

$$f(a) + f(b) = f(\phi_x(a,b)) + f(\phi_x(b,a)).$$

We will show that Test 9 is a valid tester for direct sum. First, we analyze the case that the test passes with probability 1.

▶ **Lemma 10.** *The function $f : [n]^d \to \mathbb{F}_2$ is a direct sum if and only if f passes the Diamond test with probability* 1*.*

Proof. If f is a direct sum then we can write $f(x) = \sum_i L_i(x_i)$. It follows that for any $a, b \in [n]^d$,

$$f(a) + f(b) = \sum_i (L_i(a_i) + L_i(b_i)) = f(\phi_x(a,b)) + f(\phi_x(b,a)).$$

For the other direction, suppose f passes the Diamond test with probability 1. Then, for all $x \in \mathbb{F}_2^d, a \in [n]^d$ we have

$$f(a) = f(\mathbb{1}) + f(\phi_x(\mathbb{1}, a)) + f(\phi_x(a, \mathbb{1})). \tag{1}$$

Let $e_{i,j} \in [n]^d$ denote a vector which is 1 at all coordinates except coordinate i, where it has value j. Then, repeated application of Equation (1) lets us decompose $f(a)$ as

$$f(a) = f(\mathbb{1}) + \sum_{i=1}^{d} \left(f(e_{i,a_i}) + f(\mathbb{1})\right).$$

Thus, f is a direct sum. ◀

As a simple corollary of Lemma 10 we have:

▶ **Corollary 11.** *If f is ε-close to a direct sum, then f passes the Diamond test with probability at least $1 - 4\varepsilon$.*

Proof. Let g be a direct sum with $\mathsf{dist}(f, g) \leq \varepsilon$. By a union bound, f and g agree on all 4 points queried by the test (because each query is marginally uniformly distributed), in which case the Diamond test accepts. ◀

With completeness done, the remainder of this section is focused on showing the Diamond test's soundness as stated in Theorem 4. We restate the theorem below.

▶ **Theorem 12.** *Suppose $f : [n]^d \to \mathbb{F}_2$ passes the Diamond test with probability $1 - \varepsilon$. Then, f is $C_n\varepsilon$-close to a direct sum, for some constant C_n independent of d.*

We sketch the ideas used in the analysis. First, we will generalize to a "4-function" version of the test, which is more amenable to induction. Then, we use hypercontractivity ([1, 19, 3]) to prove Theorem 17, which establishes a dichotomy for 4-tuples of functions that pass the test with probability $1 - \varepsilon$: if (f, g, k, h) passes the test with probability ε, then either f (and g, h, k as well) is 2ε-close to a direct sum, or f is $\Omega_n(1)$-far from a direct sum. Next, we will relate the pass probability of (f, g, h, k) to the pass probability of 1-variable restrictions of f, g, k, h. This allows us to go down one dimension and apply induction. Crucially, the dichotomy theorem allows us to avoid our closeness factor deteriorating at each step of the induction, which would result in a dependence on the dimension in our result. This dichotomy based approach is inspired by the work of [7, 10], but the details of establishing the dichotomy and performing the induction are quite different.

3.1 The Four Function Diamond Test

In order to facilitate our inductive approach we define a four function version of the diamond test.

▶ **Test 13** (4-Function Diamond)**.** Sample $x \sim \mathbb{F}_2^d$ and $a, b \sim [n]^d$. Accept iff

$$f(a) + g(\phi_x(a, b)) + h(\phi_x(b, a)) + k(b) = 0.$$

The following lemma characterizes what happens when four functions pass with probability 1. It will be used later to establish the base case (in constant dimensions) of our inductive analysis.

▶ **Lemma 14.** *Suppose (f, g, h, k) passes the 4-function Diamond test with probability 1. Then f, g, k, h are direct sums.*

Proof. If (f, g, h, k) passes the 4-function Diamond test with probability 1 then for all a, x, b we have:

$$f(a) + g(\phi_x(a, b)) + h(\phi_x(b, a)) + k(b) = 0 = f(b) + g(\phi_{x+\mathbb{1}}(b, a)) + h(\phi_{x+\mathbb{1}}(a, b)) + k(a).$$

Using Fact 8 to simplify we get that for all a, b,

$$f(a) + k(a) = f(b) + k(b).$$

Again using the fact that (f, g, h, k) passes the test with probability 1 we have that for all a, x, b:

$$f(a) + g(\phi_x(a, b)) + h(\phi_x(b, a)) + k(b) = 0 = f(a) + g(\phi_{x+\mathbb{1}}(a, b)) + h(\phi_{x+\mathbb{1}}(b, a)) + k(b).$$

By Fact 8 this implies

$$g(\phi_x(a,b)) + g(\phi_x(b,a)) = h(\phi_x(a,b)) + h(\phi_x(b,a)).$$

Letting $c = \phi_x(a,b), d = \phi_x(b,a)$, and noting that c, d can be arbitrary (unrelated) elements of $[n]^d$, we have that for all c, d.

$$g(c) + g(d) = h(c) + h(d).$$

Combining our observations so far we conclude that there exist constants α, β such that for all $a \in [n]^d$ we have

$$f(a) = k(a) + \alpha, \quad g(a) = h(a) + \beta.$$

Thus, for all a, b we have

$$f(a) + f(b) = g(\phi_x(a,b)) + g(\phi_x(b,a)) + \alpha + \beta.$$

Taking $x = 0$ we conclude

$$f(a) + f(b) = g(a) + g(b) + \alpha + \beta.$$

Thus, there exists γ such that for all a we have

$$f(a) = g(a) + \gamma.$$

In summary, we have shown that for all a, b, x, we have

$$f(a) + f(b) = f(\phi_x(a,b)) + f(\phi_x(b,a)) + \alpha + \beta.$$

Setting $x = 0$ again shows that $\alpha + \beta = 0$. Thus, we have that f passes the (1-function) Diamond test with probability 1. By Lemma 10 this implies that f is a direct sum. Finally, recall that we have shown in the course of our proof that g, h, k all differ from f by a constant. This concludes the proof. ◀

3.2 A Dichotomy for Functions Passing the Diamond Test

We are now prepared to show Theorem 17, which states that if the 4-function Diamond test accepts (f, g, h, k) with probability exactly $1 - \varepsilon$ then f is either 2ε-close to a direct sum, or $\Omega_n(1)$-far from being a direct sum. To this end, we first establish two lemmas which will useful in the proof.

▶ **Lemma 15.** *Fix $d \in \mathbb{N}$ and functions $f, g, h : [n]^d \to \mathbb{F}_2$. Suppose f, g, h have*

$$\Pr_{a,b \sim [n]^d, x \sim \mathbb{F}_2^d}[f(a) + g(\phi_x(a,b)) + h(\phi_x(b,a)) = 1] < \varepsilon.$$

Then, f, g, h are 3ε-close to constants.

Proof. Fix f, g, h as in the theorem. Performing a union bound and using Fact 8 we have

$$\Pr_{a,b,x}[f(a) + g(\phi_x(a,b)) + h(\phi_x(b,a)) = f(b) + g(\phi_x(a,b)) + h(\phi_x(b,a))] > 1 - 2\varepsilon.$$

Thus, f is 2ε-close to a constant α. Then, by a union bound we have

$$\Pr_{a,b,x}[g(\phi_x(a,b)) + h(\phi_x(b,a)) = \alpha + 1] > 1 - 3\varepsilon.$$

Of course, the distribution of $(\phi_x(a,b), \phi_x(b,a))$ is the same as the distribution of (a,b). Thus, we have

$$\Pr_{a,b}[g(a) + h(b) = \alpha + 1] > 1 - 3\varepsilon.$$

By the probabilistic method this implies that g, h are each 3ε-close to constants. ◀

We will also need the following lemma, which is an extension of the classic **small set expansion** property of the noisy hypercube (see [24]) to *grids*.

▶ **Fact 16** ([24]). Fix $n \in \mathbb{N}$. There exists $\lambda_n > 0$ such that for any d, and any set $S \subseteq [n]^d$ we have

$$\Pr_{a \sim [n]^d, b \sim \mathsf{T}_{1/2}(a)}[a \in S \wedge b \in S] \leq 2\mu(S)^{1+\lambda_n}.$$

Now we are ready to establish the dichotomy theorem.

▶ **Theorem 17.** *Fix $n \in \mathbb{N}$. There exists a constant $c_n > 0$ such that the following is true for any d and any functions $f, g, h, k : [n]^d \to \mathbb{F}_2$. Suppose k has $\mathsf{dist}(k, \mathsf{directsum}) < c_n$. Then, the 4-function Diamond test rejects (f, g, h, k) with probability at least $\mathsf{dist}(k, \mathsf{directsum})/2$.*

Proof. Fix f, g, k, h as described in the theorem statement. Let χ be the closest direct sum to k. Then,

$$\chi(a) + \chi(b) = \chi(\phi_x(a,b)) + \chi(\phi_x(b,a)).$$

Thus, our goal is equivalent to showing that

$$\Pr_{a,x,b}[(\chi+f)(a) + (\chi+g)(\phi_x(a,b)) + (\chi+h)(\phi_x(b,a)) \neq (\chi+k)(b)] \geq \mathsf{dist}(k, \mathsf{directsum})/2.$$

Thus, it suffices to consider the case that the closest direct sum to k is the zero function. We assume this for the remainder of the proof. Then, $\mathsf{dist}(k, \mathsf{directsum}) = \mu(k)$. We are already done unless

$$\Pr_{a,x,b}[f(a) + g(\phi_x(a,b)) + h(\phi_x(b,a)) = 1] \leq 2\mu(k). \tag{2}$$

Thus, we may assume that Equation (2) is true. Now, by Lemma 15 we have that f, g, h are $6\mu(k)$-close to constants. Without loss of generality we may assume g, h are both closer to 0 than to 1; otherwise we can add 1 to both g, h, which does not affect the sum $g(\phi_x(a,b)) + h(\phi_x(b,a))$. If f were closer to 1 than to 0 then we would have

$$\Pr_{(a,x,b)}[f(a) = 1, g(\phi_x(a,b)) = h(\phi_x(b,a)) = 0] \geq 1 - 18\mu(k) > 2\mu(k),$$

with the final inequality holding by our assumption that $\mu(k) < c_n$ (we will choose $c_n < .001$); but this would contradict Equation (2), so it cannot happen. Thus, f, g, h are all $6\mu(k)$-close to 0.

Now, we return to analyzing the probability that the 4-function Diamond test rejects (f, g, h, k); we write $\Pr[\mathsf{rej}]$ to denote the probability. Then,

$$\begin{aligned}
\Pr[\mathsf{rej}] &\geq \Pr_{a,x,b}[k(b) = 1 \wedge f(a) = g(\phi_x(a,b)) = h(\phi_x(b,a)) = 0] \\
&\geq \mu(k) - \Pr_{a,b}[k(b) = f(a) = 1] - \Pr_{a,x,b}[k(b) = g(\phi_x(a,b)) = 1] - \Pr_{a,x,b}[k(b) = h(\phi_x(b,a)) = 1] \\
&\geq \mu(k) - \mu(k)\mu(f) - \Pr_{a,b \sim \mathsf{T}_{1/2}(a)}[k(b) = g(a) = 1] - \Pr_{a,b \sim \mathsf{T}_{1/2}(a)}[k(b) = h(a) = 1].
\end{aligned}$$

Then, using Fact 16 we have

$$\Pr[\mathsf{rej}] \geq \mu(k)(1-\mu(f)) - (\mu(k)+\mu(g))^{1+\lambda_n} - (\mu(k)+\mu(h))^{1+\lambda_n}),$$

where $\lambda_n > 0$ is the constant from Fact 16. Recalling that $\mu(f), \mu(g), \mu(h) \leq 6\mu(k)$ we have:

$$\Pr[\mathsf{rej}] \geq \mu(k)(1-\mu(k)-100\mu(k)^{\lambda_n}).$$

Thus, there is some constant c_n (dependent only on n) such that if $\mu(k) \leq c_n$ then the rejection probability in is at least $\mu(k)/2$, as desired. ◀

3.3 A Lemma Concerning Restrictions

To complete our analysis of the Diamond test, we will relate the Diamond test to lower dimensional Diamond tests, and argue inductively. The Dichotomy lemma just established will be the key to ensuring that our soundness parameter does not deteriorate as we perform the induction. First, we need a lemma that controls the restrictions of a function which is far from a direct sum; namely, we show that a function which is far from a direct sum cannot have too many restrictions which are close to direct sums.

For some set $R \subseteq [n]^d$, we let $f|_R$ denote the restriction of f to R, so that $f|_R : S \to \mathbb{F}_2$ is given by $f|_R(x) = f(x)$ for all $x \in R$. We will specifically be considering restrictions where R is of the form $R = \{x \in [n]^d \mid x_i = j\}$ for some $1 \leq i \leq d$ and some $j \in [n]$. We call such R, *one-variable restrictions*. Throughout this subsection we use $\mathbb{1}$ to denote the indicator of an event. The goal of this subsection is to establish the following Lemma.

▶ **Lemma 18.** *Suppose there are $100n^2/\varepsilon$ one-variable restrictions R such that $f|_R$ is ε-close to a direct sum on R. Then, f is $O(\varepsilon)$-close to a direct sum.*

We will need Dinur and Golubev's result regarding the soundness of the square in the cube test, which we state below.

▶ **Theorem 19** ([13])**.** *Let $f : [n]^d \to \mathbb{F}_2$ satisfy,*

$$\Pr_{a,b\in[n]^d, x,y,z\in\{0,1\}^d}[f(a)+f(\phi_x(a,b))+f(\phi_y(a,b))+f(\phi_{x+y}(a,b))=0] \geq 1-\varepsilon.$$

Then f is $O(\varepsilon)$-close to a direct sum.

At a high level our proof of Lemma 18 follows the strategy of the plane versus point analysis of [25]. Suppose that there are many restrictions R_i with local direct sums $f_i : R_i \to \mathbb{F}_2$ such that f_i and f are close. We will first show that many – in fact at least a constant fraction – of these f_i's are actually consistent with each other. Using these consistent f_i's we can then define a function F on the whole space and argue that 1) F is close to f and 2) F passes the square in the cube test with high probability and is thus close to a direct sum. For both 1) and 2) we are crucially using the fact that F is defined using many consistent direct sums, so for nearly every point x in the domain, there are in fact many of these local direct sums defined on x.

3.3.1 A Sampling Lemma

In order to carry out the mentioned strategy, we must first show a sampling lemma. This lemma roughly states that given a large enough set of restrictions R_i, the measure on $[n]^d$ produced by first choosing a random R_i and then choosing a random $x \in R_i$ is nearly equal to the uniform measure up to constant factors (which will be negligible).

Let $\mathcal{R}$ be a set of one variable restrictions and let $M = |\mathcal{R}|$. Define the measure $\nu_{\mathcal{R}}$ on points $[n]^d$ via the following sampling procedure:

- Choose $R \in \mathcal{R}$.
- Choose $x \in R$.

Letting $N_{\mathcal{R}}(x) = |\{R \in \mathcal{R} \mid R \ni x\}|$, it is clear that,

$$\nu_{\mathcal{R}}(x) = \frac{N_{\mathcal{R}}(x)}{M} \cdot \frac{1}{n^{d-1}}.$$

For a subset of points $S \subseteq [n]^d$, we define

$$\nu_{\mathcal{R}}(S) = \sum_{x \in S} \nu_{\mathcal{R}}(x).$$

We also let μ denote the standard uniform measure on $[n]^d$, so for a set of points $S \subseteq [n]^d$, we have $\mu(S) = \frac{|S|}{n^d}$. At times we will also use μ to denote the uniform measure over grids of dimension other than d. It will be clear from context when this is the case and we remark that one should think of the dimension in this subsection as being arbitrary.

▶ **Lemma 20.** *Let $\mathcal{R}$ be a set of one-variable restrictions of size M. Then,*

$$\mathbb{E}_x[N_{\mathcal{R}}(x)] = \frac{M}{n} \quad \textit{and} \quad \mathsf{Var}(N_{\mathcal{R}}(x)) \leq \frac{M}{n}.$$

Proof. For the first part, we have, by linearity of expectation:

$$\mathbb{E}_x[N_{\mathcal{R}}] = \sum_{R \in \mathcal{R}} \mathbb{E}_x[\mathbb{1}(x \in R)] = \frac{M}{n}.$$

Towards the second part, we write

$$\begin{aligned}
\mathbb{E}[N_{\mathcal{R}}^2] &= \sum_{R_1, R_2 \in \mathcal{R}} \mathbb{E}_x[\mathbb{1}(x \in R_1) \cdot \mathbb{1}(x \in R_2)] \\
&= \sum_{R \in \mathcal{R}} \mathbb{E}_x[\mathbb{1}(x \in R)] + \sum_{R_1 \neq R_2} \mathbb{E}_x[\mathbb{1}(x \in R_1) \cdot \mathbb{1}(x \in R_2)] \\
&\leq \frac{M}{n} + \frac{M^2}{n^2}.
\end{aligned}$$

It follows that,

$$\mathrm{Var}(N_{\mathcal{R}}) = \left(\mathbb{E}_x[N_{\mathcal{R}}]\right)^2 - \mathbb{E}[N_{\mathcal{R}}^2] \leq \frac{M}{n}.$$ ◀

Applying Chebyshev's inequality, we get the following.

▶ **Lemma 21.** *For any $c > 0$, it holds that*

$$\Pr_x\left[\left|N_{\mathcal{R}}(x) - \frac{M}{n}\right| \geq c \cdot \frac{M}{n}\right] \leq \frac{n}{c^2 \cdot M}.$$

Proof. This lemma follows from a direct application of Chebyshev's inequality combined with the variance bound in Lemma 20. ◀

▶ **Lemma 22.** *For any set of points $S \subseteq [n]^d$ and any set $\mathcal{R}$ of one-variable restrictions of size M, we have*

$$\frac{1}{2}\left(\mu(S) - \frac{4n}{M}\right) \leq \nu_{\mathcal{R}}(S) \leq 2\mu(S) + \frac{5n}{M}.$$

Proof. For each integer i, let

$$m_i = \left| \left\{ x \in [n]^d \mid 2^i \frac{M}{n} \leq N_{\mathcal{R}}(x) < 2^{i+1} \frac{M}{n} \right\} \right|.$$

By Lemma 20 we get that

$$\frac{m_i}{n^d} \leq \frac{1}{(2^i - 1)^2 M/n}.$$

Towards the upper bound, we expand $\nu_{\mathcal{R}}(S)$ and perform a dyadic partitioning:

$$\nu_{\mathcal{R}}(S) = \sum_{x \in S} \nu_{\mathcal{R}}(x) = \sum_{x \in S} \frac{N_{\mathcal{R}}(x)}{M n^{d-1}} \leq \frac{1}{M n^{d-1}} \left(2 \cdot \frac{M}{n} \cdot |S| + \sum_{i=1}^{\infty} m_i 2^{i+1} \cdot \frac{M}{n} \right).$$

The first term in the parenthesis contributes $2\mu(\mathcal{S})$, whereas the contribution of the second summand can be upper bounded by

$$\frac{1}{n^d} \cdot \sum_{i=1}^{\infty} m_i 2^{i+1} \leq \frac{n}{M} \cdot \sum_{i=1}^{\infty} \frac{2^{i+1}}{(2^i - 1)^2} \leq \frac{5n}{M}$$

For the lower bound, we have

$$\nu_{\mathcal{R}}(S) = \sum_{x \in S} \nu_{\mathcal{R}}(x) \geq \sum_{x \in S, N_{\mathcal{R}} \geq M/(2n)} \nu_{\mathcal{R}}(x) \geq |\{x \in S \mid N_{\mathcal{R}}(x) \geq M/(2n)\}| \cdot \frac{1}{2n^d}.$$

To lower bound the last term above, we use Lemma 21 to get

$$\Pr_x \left[N_{\mathcal{R}}(x) < \frac{M}{2n} \right] \leq \frac{4n}{M}.$$

It follows that

$$|\{x \in S \mid N_{\mathcal{R}}(x) \geq M/(2n)\}| \geq n^d \left((\mu(S) - \frac{4n}{M} \right).$$

Putting everything together, we get the desired lower bound:

$$\nu_{\mathcal{R}}(S) \geq \frac{1}{2} \left(\mu(S) - \frac{4n}{M} \right).$$ ◀

3.3.2 Proof of Lemma 18

Suppose $\mathcal{R} = \{R_1, \ldots, R_M\}$ is the set of one-variable restrictions on which f is ε-close to a direct sum. For each R_i, let f_i be this direct sum, so

$$\mathsf{dist}(f_i, f|_{R_i}) \leq \varepsilon,$$

for each $1 \leq i \leq M$. By assumption $M \geq 100n^2/\varepsilon$. Note that if ε is large – say $\varepsilon \geq \frac{1}{100}$ – then the result is trivial, so for the remainder of the proof we suppose $\varepsilon < \frac{1}{100}$.

As a useful fact, we show that the set of direct sum functions has minimum distance $\frac{1}{n}$.

▶ **Lemma 23.** *For any two distinct direct sums f and g, we have*

$$\mathsf{dist}(f, g) \geq \frac{1}{n}.$$

Proof. Write $f(x) = \sum_{i=1}^{d} L_i(x_i)$ and $g(x) = \sum_{i=1}^{d} L'_i(x_i)$. First suppose for each i, $L_i - L'_i$ is constant, then since $f(x) \neq g(x)$, we have $\delta(f, g) = 1$ and we are done.

Suppose that there is some i such that $L_i - L'_i$ is not the constant function and without loss of generality, let this i be 1. Then,

$$\Pr_{x \in [n]^d}[f(x) \neq g(x)] = \mathbb{E}_{x_2,\ldots,x_d}\left[\Pr_{x_1}[L_1(x) - L'_1(x) \neq \sum_{i=2}^{d} L_i(x_i) - L'_i(x_i)]\right].$$

To conclude, note that for any $x_2, \ldots, x_d$, the inner probability on the right hand side above is at least $1/n$ since $L_1 - L'_1$ is not the constant function. ◀

We now construct the following graph $G = (V, E)$. The vertex set is $V = \{1, \ldots, M\}$ while the set of edges E consists of all pairs (i, j) such that the functions f_i and f_j are consistent:

$$f_i|_{R_i \cap R_j} = f_j|_{R_i \cap R_j}.$$

If $R_i \cap R_j$ is empty, then we also consider f_i and f_j consistent and have $(i, j) \in E$. We will first show that G contains a large clique. Call a graph is *transitive* if it is an edge disjoint union of cliques. Also define

$$\beta(G) = \max_{(i,j) \notin E} \Pr_{k \in V}[(i, k), (j, k) \in E].$$

Note that G is transitive if and only if $\beta(G) = 0$. The following lemma from [25] gives a sort of approximate version of this fact and states that if $\beta(G)$ is small then G is almost a transitive graph.

▶ **Lemma 24.** *The graph G can be made transitive by removing at most $3\sqrt{\beta(G)}|V|^2$ edges.*

In our next two lemmas we show that G has many edges, and then bound $\beta(G)$. This will establish that G contains a large clique, which corresponds to many local direct sums f_i that are consistent with one another.

▶ **Lemma 25.** *The graph G has at least $0.9M^2$ edges.*

Proof. For each R_i, let $S_i \subseteq R_i$ be the set of points on which f_i and f differ. We have that

$$\frac{|S_i|}{|R_i|} \leq \varepsilon,$$

by assumption.

Now fix an R_i. We will apply Lemma 22 inside of R_i, which is isomorphic to the grid $[n]^{d-1}$. For each j, let $R'_j = R_i \cap R_j$. Let $\mathcal{R}' = \{R'_j \mid \frac{R'_j \cap S_i}{R'_j} \geq 3\varepsilon\}$, and let $M' = |\mathcal{R}'|$. Now consider the size of S_i inside of R_i under the measure $\nu_{\mathcal{R}'}$. We have,

$$\nu_{\mathcal{R}'}(S_i) \geq 3\varepsilon.$$

By Lemma 22

$$3\varepsilon \leq \nu_{R'}(S_i) \leq 2\varepsilon + 5n/M'.$$

It follows that,

$$M' \leq \frac{5n}{\varepsilon}.$$

Thus,

$$\Pr_{R_i,R_j}\left[\frac{|R_i \cap R_j \cap S_i|}{|R_i \cap R_j|} \geq 3\varepsilon\right] \leq \frac{5n}{\varepsilon M}.$$

By a union bound, it follows that with probability at least $1 - \frac{10n}{\varepsilon M}$, we have that both

$$\frac{|R_j \cap R_i \cap S_i|}{|R_j \cap R_i|} \leq 3\varepsilon \quad \text{and} \quad \frac{|R_j \cap R_i \cap S_j|}{|R_j \cap R_i|} \leq 3\varepsilon.$$

For such i, j, we get that

$$\Pr_{x \in R_i \cap R_j}[f_i(x) = f_j(x)] \geq 1 - \Pr_{x \in R_i \cap R_j}[x \in S_i \cup S_j] \geq 1 - 6\varepsilon > \frac{1}{n},$$

and $f_i|_{R_i \cap R_j} = f_j|_{R_i \cap R_j}$ by Lemma 23. Thus choosing i, j randomly, we get that they are adjacent in G with probability at least $1 - \frac{10n}{\varepsilon M} \geq 0.9$. ◀

▶ **Lemma 26.** *Suppose R_i, R_j are distinct one-variable restrictions such that $f_i|_{R_i \cap R_j} \neq f_j|_{R_i \cap R_j}$. Then,*

$$\Pr_{R_k}[f_i|_{R_i \cap R_j \cap R_k} = f_j|_{R_i \cap R_j \cap R_k}] \leq \frac{5n^2}{M} \leq \frac{\varepsilon}{20}.$$

Proof. Let $W = R_i \cap R_j$. For each k let $R'_k = R_k \cap W$, let $f'_i = f_i|_W$, and let $f'_j = f_j|_W$.

Throughout the proof we will consider W as our ambient space and we will apply Lemma 22 inside of W. Note that we may do so because W is isomorphic to the grid $[n]^{d-2}$. Define the following set of one-variable restrictions inside W:

$$\mathcal{R}' = \{W \cap R_k \neq \emptyset \mid R_k \in \mathcal{R}, f'_i|_{R'_k} = f'_j|_{R'_k}\}.$$

We will show that $|\mathcal{R}'|$ cannot be too large. Indeed, let M' denote its size and let $S \subseteq W$ be

$$S = \{x \in W \mid f'_i(x) \neq f'_j(x)\}.$$

We have that $\frac{|S|}{|W|} \geq \frac{1}{n}$. On the other hand, by definition of $\mathcal{R}'$, we have that $\nu_{R'}(S) = 0$. Therefore by Lemma 22:

$$0 \geq \frac{1}{2}\left(\frac{1}{n} - \frac{4n}{M'}\right).$$

It follows that

$$M' \leq 4n^2.$$

Finally there can be up to $2n$ additional restrictions R_k such that $R_k \cap R_i = \emptyset$ or $R_k \cap R_j = \emptyset$. The result follows. ◀

Combining Lemma 25 and Lemma 26, we get that G contains a transitive subgraph with at least $0.8M^2$ edges. Let $\mathcal{C}_1, \ldots, \mathcal{C}_J$ be the edge disjoint union cliques of this transitive subgraph and say $\mathcal{C}_1$ is the largest one. It follows that,

$$0.8M^2 \leq \sum_{I=1}^{J} |\mathcal{C}_i|^2 \leq |\mathcal{C}_1| \cdot M,$$

and G contains a clique of size $M_1 = 0.8M$. From now we let $\mathcal{C}$ denote this clique and let $\mathcal{C} = \{1, \ldots, M_1\}$. We define the following function F using the functions $f_1, \ldots, f_{M_1}$: set $F(x) = f_j(x)$ if there is some $j \in \mathcal{C}_1$ such that $x \in R_j$, and $F(x) = 0$ otherwise.

To conclude, we will show that F is close to f and F is close to a direct sum. This will establish that f is close to a direct sum.

▶ **Lemma 27.** *F is 3ε-close to f.*

Proof. Let $S = \{x \in [n]^d \mid f(x) \neq F(x)\}$. Since f is ε-close to F on the restrictions R_i for all $i \in \mathcal{C}$, we have

$$\nu_{\mathcal{C}}(S) \leq \varepsilon.$$

By Lemma 22, it follows that,

$$\varepsilon \geq \frac{1}{2}\left(\mu(S) - \frac{4n}{0.8M}\right),$$

and

$$\mu(S) \leq 3\varepsilon.$$ ◀

▶ **Lemma 28.** *F is $O(\varepsilon)$-close to a direct sum.*

Proof. We will show that F passes the square in a cube test with high probability. Recall this test operates by choosing $a, b \in [n]^d$, $x, y \in \{0,1\}^d$ and checking if F satisfies,

$$F(a) + F(\phi_x(a,b)) + F(\phi_y(a,b)) + F(\phi_{x+y}(a,b)) = 0.$$

Note that F passes if there is some $i \in \mathcal{C}$ such that $a, b \in R_i$. Indeed, in this case all of the queried points are in R_i, so F and the direct sum f_i agree on all of the queried points and

$$F(a)+F(\phi_x(a,b))+F(\phi_y(a,b))+F(\phi_{x+y}(a,b)) = f_i(a)+f_i(\phi_x(a,b))+f_i(\phi_y(a,b))+f_i(\phi_{x+y}(a,b)) = 0.$$

To bound the probability that $i \in \mathcal{C}$ such that $a, b \in R_i$, let $N'(x) = |\{i \in \mathcal{C} \mid x \in R_i\}|$. By Lemma 21,

$$\Pr_a\left[N'(a) \geq \frac{M}{1.6n}\right] \leq \frac{5n}{M}.$$

Now condition on a satisfying $N'(a) \geq \frac{M}{1.6n}$ and let $\mathcal{C}'' = \{i \in \mathcal{C} \mid a \in R_i\}$. We label $\mathcal{C}'' = \{1, \ldots, M''\}$ where $M'' \geq \frac{M}{1.6n}$ and let $N''(x) = |\{i \in \mathcal{C}'' \mid x \in R_i\}|$. Then by Lemma 21 again,

$$\Pr_b[N''(b) \geq 0] \leq \frac{n}{M''} \leq \frac{1.6n^2}{M}.$$

By a union bound, with probability at least $1 - \frac{7n^2}{M}$ we have that a and b are both contained in some R_i with $i \in \mathcal{C}$. It follows that F passes the square in a cube test with probability at least $1 - \frac{7n^2}{M}$. By Theorem 19, F is $O(\frac{n^2}{M}) = O(\varepsilon)$-close to a direct sum. ◀

Proof of Lemma 18. The result follows by combining Lemma 27, Lemma 28 and using the triangle inequality. ◀

3.4 Soundness of the Diamond Test

Equipped with Theorem 17 we complete the proof of Theorem 4. Ultimately, we wish to conclude that if the 4-function Diamond test rejects (f, g, h, k) with probability exactly ε, then f, g, h, k are all $O(\varepsilon)$-close to direct sums. However, what we have thus far only allows us to to conclude that f, g, k, h are each *either* c_n-far from being a direct sum, *or* 2ε-close to a direct sum. Our next theorem rules out the former possibility.

▶ **Theorem 29.** *Fix $n \in \mathbb{N}$. Let c_n be the constant from Theorem 17. There is a constant δ_n such that the following holds. Fix $d \in \mathbb{N}$, and functions $f, g, h, k : [n]^d \to \mathbb{F}_2$. Suppose f is c_n-far from being a direct sum. Then, the 4-function Diamond test rejects (f, g, h, k) with probability at least δ_n.*

Proof. By Lemma 18 there exists a constant K_n such that if f has at least $100n^2/c_n$ restrictions which are c_n/K_n-close to affine, then f is c_n-close to affine. Define $N_n = \lceil 100n^2/c_n \rceil + 1$ and $\delta_n = \min(c_n/(2K_n), n^{-3N_n})$. We will prove the theorem by induction on the dimension d.

The base case is $d \leq N_n$. If $d \leq N_n$ then by Lemma 10 the 4-function Diamond test rejects (f, g, h, k) with non-zero probability since f is not a direct sum. But, any non-zero probability over events determined by a, b, x is at least $1/n^{3N_n} \geq \delta_n$.

Now, fix $d > N_n$, and assume the theorem for functions on $\mathbb{F}_2^{d-1}$. Let $f|_{i,\sigma} : [n]^{d-1} \to \mathbb{F}_2$ denote the function obtained by taking f and fixing $x_i = \sigma$. Then, there is some $i \in [d]$ such for all $\sigma \in [n]$ the restriction $f|_{i,\sigma}$ is c_n/K_n-far from a direct sum. To analyze the probability that the 4-function Diamond test rejects (f, g, h, k), we consider a partition of the probability space into $2n^2$ disjoint events. The events are the value in $[n]^2 \times \mathbb{F}_2$ that our sampled (a_i, b_i, x_i) takes. Now, consider the probability that the 4-function Diamond test rejects (f, g, h, k), conditional on $(a_i, b_i, x_i) = (\alpha, \beta, \xi)$ for some α, β, ξ. Conditional on this event, we will sample $a, b \in [n]^{d-1}, x \in \mathbb{F}_2^{d-1}$ and check:

$$f|_{i,\alpha}(a) + g|_{i,\phi_\xi(\alpha,\beta)}(\phi_x(a,b)) = h|_{i,\phi_\xi(\beta,\alpha)}(\phi_x(b,a)) + k|_{i,\beta}(b).$$

This is precisely a 4-function Diamond test on $(d-1)$-dimensional functions. Now we consider two cases.

Case 1: $f|_{i,\alpha}$ is c_n-far from being a direct sum.
Then by the inductive hypothesis this $(d-1)$-dimension test will reject with probability at least δ_n.

Case 2: $\mathsf{dist}(f|_{i,\alpha}, \mathsf{directsum}) \in [c_n/K_n, c_n]$.
Then, by Theorem 17 this $(d-1)$-dimensional test rejects with probability at least $c_n/(2K_n) \geq \delta_n$.

We have seen that in all cases, the lower dimensional 4-function Diamond tests reject with probability at least δ_n. Thus, the actual 4-function Diamond test rejects with probability δ_n as well. ◀

Combining Theorem 17 and Theorem 29 we have get the following corollary and conclude the proof of Theorem 4.

▶ **Corollary 30.** *If (f, g, h, k) passes the 4-function Diamond test with probability $1 - \varepsilon$, then f, g, h, k are each $O_n(\varepsilon)$-close to direct sums.*

Proof. The result follows immediately from Theorem 17 and Theorem 29. ◀

We can now conclude the proof of Theorem 4 and establish soundness for the Diamond test.

Proof of Theorem 4. The result follows from Corollary 30 by setting $f = g = h = k$. ◀

3.5 The Fourier Analytic Interpretation of the Diamond Test

In this section, we give a Fourier analytic interpretation of the Diamond test. Let Even denote the class of functions $f : \mathbb{F}_2^d \to \mathbb{F}_2$ satisfying $f(x + \mathbb{1}) = f(x)$ and let Odd denote the class of functions $f : \mathbb{F}_2^d \to \mathbb{F}_2$ satisfying $f(x + \mathbb{1}) = f(x) + 1$. Let $\mathsf{EvenOrOdd} = \mathsf{Even} \cup \mathsf{Odd}$. We note that $\mathsf{EvenOrOdd}$ has a nice Fourier analytic interpretation: $f \in \mathsf{EvenOrOdd}$ implies that $(-1)^f$ has either all its Fourier-mass on even-sized characters, or all of its Fourier-mass on odd-sized characters. Given a function $f : \mathbb{F}_2^d \to \mathbb{F}_2$, define random variable $\mathcal{R}(f)$ to be a random restriction of f obtained as follows: for each $i \in [d]$ independently, with probability $1/4$ restrict x_i to 0, with probability $1/4$ restrict x_i to 1, and with probability $1/2$ do not restrict x_i. We now show that the Diamond test's correctness is equivalent to the following Fourier analytic fact

▶ **Theorem 31.**

$$\mathsf{dist}(f, \mathsf{affine}) \leq O(\mathbb{E}[\mathsf{dist}(\mathcal{R}(f), \mathsf{EvenOrOdd})])$$

In fact, we show something stronger: there is a very tight quantitative relationship between

$$\mathbb{E}[\mathsf{dist}(\mathcal{R}(f), \mathsf{EvenOrOdd})]$$

and the probability that the Diamond test rejects f.

▶ **Lemma 32.** *Fix f. Let ε be the probability that the Diamond test rejects f, and let*

$$\delta = \mathbb{E}[\mathsf{dist}(\mathcal{R}(f), \mathsf{EvenOrOdd})].$$

Then,

$$2\delta \leq \varepsilon \leq 4\delta.$$

Proof. Fix f, ε, δ as in the lemma statement. Given a, b define $C_{a,b} = \left\{\phi_x(a,b) \mid x \in \mathbb{F}_2^d\right\}$ and $f_{a,b}(x) = f(\phi_x(a,b))$. We have:

$$\varepsilon = \Pr_{a,b,x}[f_{a,b}(x) + f_{a,b}(x + \mathbb{1}) \neq f_{a,b}(0) + f_{a,b}(\mathbb{1})]. \tag{3}$$

▷ Claim 33. $\varepsilon \leq 4\delta$.

Let $\delta_{a,b} = \mathsf{dist}(f_{a,b}, \mathsf{EvenOrOdd})$. By a union bound we have

$$\Pr_{z,x}[f_{a,b}(x + \mathbb{1}) + f_{a,b}(x) \neq f_{a,b}(z + \mathbb{1}) + f_{a,b}(z)] \leq 4\delta_{a,b}.$$

By averaging, $\delta = \mathbb{E}_{a,b}[\delta_{a,b}]$. Thus,

$$\Pr_{a,b,z,x}[f_{a,b}(x + \mathbb{1}) + f_{a,b}(x) \neq f_{a,b}(z + \mathbb{1}) + f_{a,b}(z)] \leq 4\delta. \tag{4}$$

Now, observe that for random a, b, z, x the following two distributions are the same:
1. $(\phi_x(a,b), \phi_x(b,a), a, b)$
2. $(\phi_x(a,b), \phi_x(b,a), \phi_z(a,b), \phi_z(b,a))$.

Thus Equation (4) implies $\varepsilon \leq 4\delta$.

▷ Claim 34. $\varepsilon \geq 2\delta$.

For any function g,

$$\Pr_x[g(x) \neq g(x+\mathbb{1})] = 2\mathsf{dist}(g, \mathsf{Even}),$$

$$\Pr_x[g(x) = g(x+\mathbb{1})] = 2\mathsf{dist}(g, \mathsf{Odd}).$$

Thus, for any a, b we have

$$\Pr_x[f_{a,b}(x) + f_{a,b}(x+\mathbb{1}) \neq f_{a,b}(0) + f_{a,b}(\mathbb{1})] \geq 2\mathsf{dist}(f_{a,b}, \mathsf{EvenOrOdd}).$$

Averaging over a, b on both sides gives $\varepsilon \geq 2\delta$. ◀

Proof of Theorem 31. The result follows from the soundness of the Diamond Test, Theorem 4 ◀

4 The Affine on Subcube Test

In this section we consider a class of direct sum tests that generalizes the Square in Cube test of [13]. We call these tests **Affine on Subcube** tests. Indeed, the Square in Cube test can be viewed as performing the BLR affinity test inside of a random induced hypercube inside of $[n]^d$. We show that in fact testing for affinity on a random subcube with *any* affinity tester also works as a direct sum test.

▶ **Test 35** (Affine on Subcube)**.** Sample $a, b \in [n]^d$ and run an affinity test that has soundness function $s(\delta) = \Omega(\delta)$ on the function $x \mapsto f(\phi_x(a, b))$.

The flexibility of being able to use any affinity tester also enables us to obtain a direct sum test with additional properties such as erasure resiliency, which we discuss further in Section 4.2, and less use of randomness, which we discuss in Section 4.1. Furthermore, we believe that our analysis for going from local direct sums to a global direct sum is simpler than that of [13] and thus noteworthy on its own.

One final benefit of our analysis is that it fixes an error in [13]'s analysis. Specifically, they establish that a function f passing the Square in Cube test with probability $1 - \varepsilon$ is *locally* close to a direct sum (this is basically Lemma 39 in our presentation), but they do not show how to combine these local direct sums into a global direct sum.

We now show that the Affine on Subcube test is a valid direct sum tester. Once again, completeness is clear. If f is a direct sum, then the Affine on Subcube test passes with probability 1. Our focus is then on showing soundness.

▶ **Theorem 36.** *Fix $f : [n]^d \to \mathbb{F}_2, \varepsilon \geq 0$ and an affinity test T. If f passes the Affine on Subcube test (instantiated with T) with probability $1 - \varepsilon$, then f is $C\varepsilon$-close to a direct sum, for some constant C independent of n, d.*

Proof. Fix f as in the theorem statement. That is, satisfying

$$\mathbb{E}_{a,b\sim[n]^d}[\mathsf{dist}(x \mapsto f(\phi_x(a, b)), \mathsf{affine})] \leq O(\varepsilon). \tag{5}$$

We re-interpret Equation (5) in three steps. First, observe that Equation (5) is equivalent to

$$\mathbb{E}_{a,b\sim[n]^d, z\sim\mathbb{F}_2^d}[\mathsf{dist}(x \mapsto f(\phi_z(a, b)) + f(\phi_{x+z}(a, b)), \mathsf{linear})] \leq O(\varepsilon). \tag{6}$$

Next, observe that we can rewrite $\phi_{x+z}(a, b)$ as follows:

$$\phi_{x+z}(a, b) = \phi_x(\phi_z(a, b), \phi_z(b, a)).$$

In light of this observation, Equation (6) is equivalent to

$$\mathbb{E}_{a,b\sim[n]^d,z\sim\mathbb{F}_2^d}[\mathsf{dist}(x\mapsto f(\phi_z(a,b))+f(\phi_x(\phi_z(a,b),\phi_z(b,a))),\mathsf{linear})]\leq O(\varepsilon). \tag{7}$$

Of course, the distribution of $(\phi_z(a,b),\phi_z(b,a))$ is the same as the distribution of (a,b). So, we can rewrite Equation (7) as:

$$\mathbb{E}_{a,b\sim[n]^d}[\mathsf{dist}(x\mapsto f(a)+f(\phi_x(a,b)),\mathsf{linear})]\leq O(\varepsilon). \tag{8}$$

This interpretation of the test is more convenient to work with.

For each a,b define f_{ab} to be the function $x\mapsto f(a)+f(\phi_x(a,b))$, and let ε_{ab} be the probability that f_{ab} fails the linearity test. Clearly $\mathbb{E}[\varepsilon_{ab}]=\mathbb{E}[\varepsilon_a]\leq O(\varepsilon)$. Because f_{ab} passes the linearity test with probability $1-\varepsilon_{ab}$, there is vector $F^a(b)\in\mathbb{F}_2^d$ such that

$$\Pr_x[f_{ab}(x)=\langle F^a(b),x\rangle]>1-O(\varepsilon_{ab}), \tag{9}$$

where $\langle v,w\rangle=\sum_i v_iw_i$. We will now show that $b\mapsto F^a(b)$ is close to a direct product, by proving that this function passes a certain direct product test. Dinur and Golubev [13] give the following direct product test (which is slightly more convenient here than the original direct product test of [16]):

▶ **Test 37** (Direct Product Test)**.** Sample $x\in[n]^d$. Sample a set A by including each $i\in[d]$ in A with probability $3/4$. Sample y by setting $y_i=x_i$ if $i\in A$, and otherwise sampling y_i uniformly from $[n_i]$. Accept iff $f(x)_i=f(y)_i$ for all $i\in A$.

Using [11], Dinur and Golubev show

▶ **Fact 38.** If f passes Test 37 with probability $1-\varepsilon$, then f is $O(\varepsilon)$-close to a direct product.

Now, we use Fact 38 to show that $b\mapsto F^a(b)$ is close to a direct product.

▶ **Lemma 39.** *$b\mapsto F^a(b)$ is $O(\varepsilon_a)$-close to a direct product.*

Proof. Given b,b' define $\mathcal{D}_{bb'}$ on $\mathbb{F}_2^d$ as follows. If $x\sim\mathcal{D}_{bb'}$ then for each $i\in[d]$ independently, x_i is determined as follows:

If $b_i\neq b'_i$, then $x_i=0$. If $b_i=b'_i$, then $x_i=1$ with probability $2/3$ and 0 otherwise.

Define $\mathsf{eq}_{3/4}=\mathsf{T}_{(3n/4-1)/(n-1)}$. If $b'\sim\mathsf{eq}_{3/4}(b)$, then for each i independently, $b_i=b'_i$ with probability $3/4$. Now, observe that if we sample $b'\sim\mathsf{eq}_{3/4}(b)$ and $x\sim\mathcal{D}_{bb'}$, then the values x_i are independent uniformly random bits. However, if we choose b',x in this manner, then the joint distribution of (b,b',x) satisfies

$$\phi_x(a,b)=\phi_x(a,b').$$

This is because we select a_i whenever $b_i\neq b'_i$. Then, performing a union bound on Equation (9) (and averaging away the dependence on b) we have that

$$\Pr_{b\sim[n]^d,b'\sim\mathsf{eq}_{3/4}(b),x\sim\mathcal{D}_{bb'}}[\langle F^a(b),x\rangle=f_{ab}(x)=f_{ab'}(x)=\langle F^a(b'),x\rangle]>1-O(\varepsilon_a). \tag{10}$$

For each a,b,b', define

$$\delta_{abb'}=\Pr_{x\sim\mathcal{D}_{bb'}}[\langle F^a(b)+F^a(b'),x\rangle\neq 0].$$

By Equation (10), $\mathbb{E}_{b\sim[n]^d,b'\sim\mathsf{eq}_{3/4}(b)}[\delta_{abb'}]<O(\varepsilon_a)$. Intuitively, if $\delta_{abb'}$ is small then $F^a(b),F^a(b')$ must have some "agreement". More precisely we have:

▷ Claim 40. If $\delta_{abb'} < 1/3$, then for all i where $b_i = b'_i$ we have $F^a(b)_i = F^a(b')_i$

Proof. Suppose to the contrary that there is some $i \in [d]$ with $b_i = b'_i$ but $F^a(b)_i \neq F^a(b')_i$. If $x \sim \mathcal{D}_{bb'}$ then there is a $1/3$ chance of $x_i = 0$ and a $2/3$ chance of $x_i = 1$, and this is independent of the values of x_j for $j \neq i$. This means that $\langle F^a(b) + F^a(b'), x\rangle$ takes on either value in $\mathbb{F}_2$ with probability at least $1/3$, implying that $\delta_{abb'} \geq 1/3$. ◁

Now, by Markov's Inequality we have

$$\Pr_{b\sim[n]^d, b'\sim \mathsf{eq}_{3/4}(b)}[\delta_{abb'} \geq 1/3] \leq 3\mathbb{E}_{b,b'}[\delta_{abb'}] < O(\varepsilon_a). \tag{11}$$

Given b, b' let $\overline{\Delta}(b, b')$ denote the set of coordinates i where $b_i = b'_i$. By Equation (11) we have

$$\Pr_{b\sim[n]^d, b'\sim \mathsf{eq}_{3/4}(b)}[F^a(b)_i = F^a(b')_i \quad \forall i \in \overline{\Delta}(b, b')] > 1 - O(\varepsilon_a).$$

By Fact 38 we conclude that $F^a(b)$ is $O(\varepsilon_a)$-close to a direct product. ◀

Now we conclude the proof of Theorem 36. For each a, let $(g_1(a, b_1), \ldots, g_d(a, b_d))$ be a direct product that is close $O(\varepsilon_a)$-close to $F^a(b)$; the existence of these direct products was shown in Lemma 39. Then,

$$\Pr_{a,x,b}\left[f_{ab}(x) = \sum_i g_i(a, b_i)x_i\right] > 1 - O(\varepsilon). \tag{12}$$

Observe that $\phi_x(a, b) = \phi_{\mathbb{1}+x}(b, a)$. Thus, by Equation (12) and a union bound we have

$$\Pr_{a,b,x}\left[\sum_i g_i(a, b_i)x_i + f(a) = f_{ab}(x) = f_{ba}(x + \mathbb{1}) = f(b) + \sum_i g_i(b, a_i)(1 + x_i)\right] > 1-O(\varepsilon). \tag{13}$$

For each a, b, define

$$\lambda_{ab} = \Pr_x\left[\sum_i (g_i(a, b_i) + g_i(b, a_i))x_i \neq \sum_i g_i(b, a_i) + f(a) + f(b)\right].$$

By Equation (13), $\mathbb{E}_{ab}[\lambda_{ab}] \leq O(\varepsilon)$. If $\lambda_{ab} < 1/2$ then we must have

$$\sum_i g_i(b, a_i) = f(a) + f(b).$$

By Markov's inequality:

$$\Pr_{a,b}[\lambda_{ab} \geq 1/2] \leq 2\mathbb{E}_{a,b}[\lambda_{ab}] \leq O(\varepsilon).$$

Thus, we have:

$$\Pr_{a,b}\left[f(a) = f(b) + \sum_i g_i(b, a_i)\right] > 1 - O(\varepsilon).$$

Then, by the probabilistic method there exists β such that

$$\Pr_a\left[f(a) = f(\beta) + \sum_i g_i(\beta, a_i)\right] > 1 - O(\varepsilon).$$

That is, f is $O(\varepsilon)$-close to the direct sum $a \mapsto f(\beta) + \sum_i g_i(\beta, a_i)$. ◀

4.1 Derandomization

In this section we show how to use Theorem 36 to obtain a more randomness-efficient direct sum tester than the Square in Cube test or Diamond test. In particular, we define the **Diamond in Cube test** as follows: run the Diamond test on the restriction of f to a random subcube (i.e., $f(\phi_x(a,b))$).

We now show that the Direct-Sum Diamond test is more randomness-efficient than the Square in Cube test and the Diamond in Cube test for some parameter regimes. In the following discussion we assume that n is a power of two for convenience.

In the statement of tests like the Square in Cube test we say "sample $a, b \sim [n]^d, x, y \sim \mathbb{F}_2$". This would seem to indicate that the test requires $(2+2\log_2 n)d$ bits of randomness. However, if we look inside the Square in Cube test, we see that this is not the case. The Square in Cube test actually requires sampling from the following distribution:

$$(\phi_0(a,b), \phi_x(a,b), \phi_y(a,b), \phi_{x+y}(a,b)),$$

where $a, b \sim [n]^d$ and $x \sim \mathbb{F}_2^d$. However, sampling from this distribution does not actually require sampling a, b, x, y fully. Instead, we can sample from the distribution as follows. First, sample $a \sim [n]^d, x, y \sim \mathbb{F}_2^d$. Then we partially sample b. Note that we don't need to sample b_i if $x_i = y_i = 0$, because in this case the value of b_i will not be needed in any of the query points. Thus, the average number of random bits required to create a sample for the Square in Cube test is only $(2 + 1.75\log_2 n)d$.

On the surface the Diamond test might appear to be more efficient. Rather than sampling a, b, x, y it only needs to sample a, b, x. However, the Diamond test actually requires more random bits than the Square in Cube test if $n \geq 20$. The number of random bits required to sample from the Diamond in Cube's distribution is $(2\log_2 n + (1 - 1/n))d$.

Now, we claim that the Diamond in Cube test is more randomness-efficient than both the Diamond test and the Square in Cube test, at least for large n. A simple computation shows that we can obtain samples from the Diamond in Cube test's sample distribution using only $(2.5 + 1.5\log_2 n)d$ random bits, vindicating this claim.

4.2 Direct Sum Testing in the Online Adversary Model

In this section we show that an immediate application of the Affine on Subcube test from Theorem 36 yields direct sum testers that are online erasure-resilient. The t-online erasure model was first considered by Kalemaj, Raskhodnikova, and Varma in [20] and was further studied in [22, 6]. The properties that these works consider include linearity, low degree, and various properties of sequences. Let us now describe the model.

We are once again given query access to a function $f : [n]^d \to \mathbb{F}_2$, and the goal is the same as in direct sum testing – to distinguish whether f is a direct sum, or far from a direct sum. The twist is that in the t-online erasure model, there is an adversary who is allowed to erase any t entries of f after each query that the property testing algorithm makes. If a point is queried after it is erased, then the oracle will return $\perp$ instead of the actual value $f(x)$. It is not hard to see that neither the Diamond test that we analyze nor the Square in the Cube test of [13] is erasure resilient. The reason is that in both tests, after the first three queries are made, the adversary knows what the fourth query will be and can erase f at that point.

However, by using the Affine on Subcube test with an erasure resilient affinity tester, we obtain an erasure resilient direct sum tester.

▶ **Theorem 41.** *Fix a distance parameter $\delta > 0$ and erasure parameter t, and take d sufficiently large relative to n, δ, t. Then there is a direct sum tester for $f : [n]^d \to \mathbb{F}_2$ that makes $O(\max(1/\varepsilon, \log t))$-queries in the t-online erasure model that satisfies:*

- ***Completeness:*** *If f is a direct sum, the tester accepts with probability 1.*
- ***Soundness:*** *If f is δ-far from being a direct sum, then the tester rejects with probability $2/3$.*

Proof. The result follows by combining Theorem 36 with Theorem 1.1 of [6]. Specifically we use the Affine on Subcube test where the inner affinity tester is the erasure resilient one of [6]. Technically, Theorem 1.1 of [6] is stated for linearity testing, but it is straightforward to modify their result to obtain an affinity tester. ◀

5 Reconstruction for Direct Sums

In this section we address another question raised in [13] on whether a test, which they call the **Shapka Test** can be used to reconstruct the underlying direct sum. We show that the this is indeed the case in some parameter regimes (i.e with the input function being sufficiently close to a direct sum), and then we give an alternate method for obtaining corrected versions of the direct sum, that is both more query efficient and more error tolerant than the Shapka test. We also give upper bounds that match up to a constant for the fraction of errors that can be tolerated when reconstructing the direct sum. This shows that the new reconstruction method we propose is essentially optimal: it works for essentially the entire range of ε where recovery is information theoretically possible, and there is no asymptotically more query-efficient correction method.

We start with our lower bounds.

▶ **Proposition 42.** *Fix an even n, arbitrary integer d, and some $\epsilon > 0$ such that $n\varepsilon > 1/2$. There exist distinct direct sums $L, Z : [n]^d \to \mathbb{F}_2$, and a function $f : [n]^d \to \mathbb{F}_2$ such that $\mathsf{dist}(f, L) = \mathsf{dist}(f, Z) \leq \varepsilon$.*

Proof. Let Z be the zero function, and let L be the indicator of $x_d = 1$. Let f be any function which is zero if $x_d \neq 1$, and for inputs with $x_d = 1$, f is zero half the time. Then, $\mathsf{dist}(f, L) = \mathsf{dist}(f, Z) = 1/(2n) \leq \varepsilon$. In other words, there is no unique closest direct sum to f, so decoding $f(\mathbb{1})$ is undefined. ◀

▶ **Proposition 43.** *Fix an n divisible by 4, an arbitrary integer d, and $\epsilon > 0$ such that $(0.7)^{d/n} < \varepsilon, n\varepsilon < 1/2$. Then there is a set $\mathcal{F}$ of functions $f : [n]^d \to \mathbb{F}_2$, such that using fewer than $n/4$ queries it is impossible to determine the value of $L(\mathbb{1})$ with probability greater than $1/2$, where L is the closest direct sum to f.*

Proof. Call a point $x \in [n]^d$ **heavy** if more than $2d/n$ coordinates $i \in [d]$ have $x_i = 1$. By a Chernoff bound, the fraction of points in $[n]^d$ which are heavy is less than $(0.7)^{d/n}$. Let L be a uniformly random direct sum. Define $\mathcal{F}$ to be the class of all functions which agree with L on non-heavy inputs. Then, querying $f \in \mathcal{F}$ on a heavy point is useless, because it outputs an arbitrary value unrelated to L. By our assumption that $(0.7)^{d/n} < \varepsilon$ we have that any $f \in \mathcal{F}$ is ε-close to L. Now we argue that $n/4$ non-heavy queries do not suffice to determine $L(\mathbb{1})$ with non-trivial probability. Indeed, there will be a coordinate $i \in [d]$ such that among the $n/4$ non-heavy queries, no queried point x has $x_i = 1$. Thus, we cannot hope to distinguish between the equally likely cases $L_i(1) = 0$ and $L_i(1) = 1$, and thus cannot predict $L(\mathbb{1})$ with non-trivial probability. ◀

Now we give algorithms for reconstructing values from the closest direct sum to a function f. Let us start with the **Shapka Test** of [13] which we now describe. This test is slightly different for odd and even d; for simplicity we only consider the case of odd d. Let $e_i \in \{0,1\}^d$ denote a vector with zeros except at the i-th coordinate. We describe how the Shapka Test can be used to reconstruct a direct sum.

▶ **Test 44** (Shapka [13])**.** Sample $a, b \in [n]^d$. Accept iff

$$f(b) = \sum_{i=1}^{d} f(\phi_{e_i}(a,b)). \tag{14}$$

Thus, to reconstruct a direct sum from a corrupted version, f, we can define,

$$f_{\mathsf{shapka}}(b) = \mathsf{maj}_{a\in[n]^d} \sum_{i=1}^{d} f(\phi_{e_i}(a,b)).$$

Dinur and Golubev show that if f passes the Shapka test with probability ε, then f is $O(n\varepsilon)$-close to a direct sum. Now, instead of checking Equation (14), we use the expression on the right hand side of Equation (14) as our guess for the value of $f(b)$, and take the most common value at each b to define the reconstructed function f_{shapka}. We show that if f is close enough to a direct sum, then f_{shapka} is in fact this direct sum:

▶ **Proposition 45.** *Fix $\varepsilon > 0, n \in \mathbb{N}$, and d odd with $n\varepsilon d < 1/4$. Suppose $f : [n]^d \to \mathbb{F}_2$ is ε-close to the direct sum $L = \sum_i L_i$. Then, for any b,*

$$\Pr_{a\in[n]^d}[f(b) = L(b)] \geq \frac{3}{4}.$$

It follows that $f_{\mathsf{shapka}} = L$.

Proof. We say that a query $f(x)$ **succeeds** if $f(x) = L(x)$. Let

$$\varepsilon_i = \Pr_x[f(x) \neq L(x) \mid x_i = b_i].$$

By a union bound, the Shapka test succeeds with probability at least $1 - \sum_i \varepsilon_i$. For each i,

$$\varepsilon = \Pr[f(x) \neq L(x)] \geq \Pr[f(x) \neq L(x) \wedge x_i = b_i] = \varepsilon_i/n.$$

Thus, $\sum_i \varepsilon_i \leq n\varepsilon d \leq 1/4$, so the Shapka test produces the correct value with probability at least $3/4$. ◀

Now we give a more query efficient method for reconstructing a direct sum. Our method has asymptotically optimal query complexity (by Proposition 43), and still works at essentially all possible values of ε for which the function is guaranteed to be information theoretically recoverable (by Proposition 42). We reconstruct the direct sum via the following voting scheme, where we call the reconstructed function f_{recon}. To make things more convenient, we assume that d is odd. Set

$$f_{\mathsf{recon}}(x) = \mathsf{maj}_{x^{(1)},\ldots,x^{(n)}} \sum_{i=1}^{n} x^{(i)},$$

where the set of queried points $x^{(1)}, \ldots, x^{(n)}$ is weighted in the majority according to the following distribution.

- Randomly partition $[d]$ into n parts, $S_1 \sqcup \cdots \sqcup S_n = [d]$, by choosing putting each $i \in [d]$ independently and uniformly at random in a part S_i.
- Sample R uniformly from $([n] \setminus \{1\})^d$.
- For $i \in [n]$, form query point $x^{(i)} \in [n]^d$ by setting $x^{(i)}_j = 1$ for each $j \in S_i$, and setting $x^{(i)}_j = R_j$ otherwise.

For even, n, we require $n+1$ queries, and use the following voting scheme.

$$f_{\mathsf{recon}}(x) = \mathsf{maj}_{x^{(1)},\ldots,x^{(n)}} \sum_{i=1}^{n} x^{(i)},$$

where the set of queried points $x^{(1)}, \ldots, x^{(n+1)}$ is weighted in the majority according to the following distribution.

- Randomly partition $[d]$ into $n+1$ parts, $S_1 \sqcup \cdots \sqcup S_{n+1} = [d]$, by choosing putting each $i \in [d]$ independently and uniformly at random in a part S_i.
- Sample $R \in [n]^d$ as follows: for each i independently, set $R_i = 1$ with probability $1/n^2$ and otherwise sample R_i uniformly from $[n] \setminus \{1\}$.
- For $i \in [n+1]$, form query point $x^{(i)} \in [n]^d$ by setting $x^{(i)}_j = 1$ for each $j \in S_i$, and setting $x^{(i)}_j = R_j$ otherwise.

We prove that the scheme above correctly reconstructs the underlying direct sum for the case where n is odd, as the case where n is even is similar.

▶ **Proposition 46.** *Fix n, d, ε. Let $f : [n]^d \to \mathbb{F}_2$ be ε close to a direct sum L, with $(n+1)\varepsilon < 1/4$. Then for any point x*

$$\Pr_{x^{(1)},\ldots,x^{(n)}} \left[\sum_{i=1}^{n} f(x^{(i)}) = L(x) \right] \geq \frac{3}{4}.$$

It follows that $f_{\mathsf{recon}} = L$.

Proof. Without loss of generality, let us assume that $x = \mathbb{1}$ – the all ones vector. We again say that a query $f(x)$ **succeeds** if $f(x) = L(x)$.

We claim that $x^{(1)}, \ldots, x^{(n)}$ are uniformly random points in $[n]^d$. Fix $i \in [n]$. For each $j \in [d]$, $x^{(i)}_j$ has a $1/n$ chance of being a 1, and a $1/n$ chance of being j for any $j \in [n] \setminus \{1\}$. Furthermore, the value of $x^{(i)}_j$ is independent of the value of $x^{(i)}_{j'}$ for $j' \neq j$. Thus, the probability that all n queries all succeed is at least $1 - n\varepsilon \geq 3/4$ by a union bound. If the queries all succeed, then because d is odd, we have:

$$\sum_i f(x^{(i)}) = \sum_i L(x^{(i)}) = (d-1)L(R) + L(\mathbb{1}) = L(\mathbb{1}). \qquad \blacktriangleleft$$

References

1 Rudolf Ahlswede and Peter Gács. Spreading of sets in product spaces and hypercontraction of the markov operator. *The annals of probability*, pages 925–939, 1976.

2 Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing low-degree polynomials over gf (2). In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 188–199. Springer, 2003. doi:10.1007/978-3-540-45198-3_17.

3 Prashanth Amireddy, Srikanth Srinivasan, and Madhu Sudan. Low-degree testing over grids. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2023)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.APPROX/RANDOM.2023.41.

4 Mitali Bafna, Dor Minzer, and Nikhil Vyas. Quasi-linear size pcps with small soundness from HDX. *CoRR*, abs/2407.12762, 2024. doi:10.48550/arXiv.2407.12762.

5 Mitali Bafna, Srikanth Srinivasan, and Madhu Sudan. Local decoding and testing of polynomials over grids. *Random Struct. Algorithms*, 57(3):658–694, 2020. doi:10.1002/RSA.20933.

6 Omri Ben-Eliezer, Esty Kelman, Uri Meir, and Sofya Raskhodnikova. Property testing with online adversaries. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference, ITCS 2024, January 30 to February 2, 2024, Berkeley, CA, USA*, volume 287 of *LIPIcs*, pages 11:1–11:25. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.ITCS.2024.11.

7 Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman. Optimal testing of reed-muller codes. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 488–497. IEEE, 2010. doi:10.1109/FOCS.2010.54.

8 Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 73–83, 1990. doi:10.1145/100216.100225.

9 Siu On Chan. Approximation resistance from pairwise-independent subgroups. *J. ACM*, 63(3):27:1–27:32, 2016. doi:10.1145/2873054.

10 Roee David, Irit Dinur, Elazar Goldenberg, Guy Kindler, and Igor Shinkar. Direct sum testing. In *Proceedings of the 2015 Conference on innovations in theoretical computer science*, pages 327–336, 2015. doi:10.1145/2688073.2688078.

11 Yotam Dikstein and Irit Dinur. Agreement testing theorems on layered set systems. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1495–1524. IEEE, 2019. doi:10.1109/FOCS.2019.00088.

12 Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007. doi:10.1145/1236457.1236459.

13 Irit Dinur and Konstantin Golubev. Direct Sum Testing: The General Case. In Dimitris Achlioptas and László A. Végh, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*, volume 145 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 40:1–40:11, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.APPROX-RANDOM.2019.40.

14 Irit Dinur and Or Meir. Derandomized parallel repetition via structured pcps. *Electron. Colloquium Comput. Complex.*, TR10-107, 2010. URL: https://eccc.weizmann.ac.il/report/2010/107, arXiv:TR10-107.

15 Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the pcp-theorem. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 155–164. IEEE Computer Society, 2004. doi:10.1109/FOCS.2004.16.

16 Irit Dinur and David Steurer. Direct product testing. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 188–196. IEEE, 2014. doi:10.1109/CCC.2014.27.

17 Oded Goldreich and Shmuel Safra. A combinatorial consistency lemma with application to proving the PCP theorem. *SIAM J. Comput.*, 29(4):1132–1154, 2000. doi:10.1137/S0097539797315744.

18 Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. New direct-product testers and 2-query pcps. *SIAM J. Comput.*, 41(6):1722–1768, 2012. doi:10.1137/09077299X.

19 Jeff Kahn, Gil Kalai, and Nathan Linial. *The influence of variables on Boolean functions*. Institute for Mathematical Studies in the Social Sciences, 1989.

20 Iden Kalemaj, Sofya Raskhodnikova, and Nithin Varma. Sublinear-time computation in the presence of online erasures. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 90:1–90:25. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ITCS.2022.90.

21 Tali Kaufman and Madhu Sudan. Algebraic property testing: the role of invariance. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 403–412, 2008. doi:10.1145/1374376.1374434.

22 Dor Minzer and Kai Zhe Zheng. Adversarial low degree testing. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 4395–4409. SIAM, 2024. doi:10.1137/1.9781611977912.154.

23 Dor Minzer and Kai Zhe Zheng. Near optimal alphabet-soundness tradeoff pcps. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 15–23. ACM, 2024. doi:10.1145/3618260.3649606.

24 Ryan O'Donnell. Analysis of boolean functions, 2021. arXiv:2105.10386.

25 Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 475–484. ACM, 1997. doi:10.1145/258533.258641.

Toward Separating QMA from QCMA with a Classical Oracle

Mark Zhandry
NTT Research, Sunnyvale, CA, USA

Abstract

QMA is the class of languages that can be decided by an efficient quantum verifier given a *quantum* witness, whereas QCMA is the class of such languages where the efficient quantum verifier only is given a *classical* witness. A challenging fundamental goal in quantum query complexity is to find a classical oracle separation for these classes. In this work, we offer a new approach towards proving such a separation that is qualitatively different than prior work, and show that our approach is sound assuming a natural statistical conjecture which may have other applications to quantum query complexity lower bounds.

2012 ACM Subject Classification Theory of computation → Quantum complexity theory

Keywords and phrases Quantum, Oracle Separations, QMA, QCMA

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.95

Related Version *Full Version*: https://arxiv.org/abs/2411.01718

1 Introduction

Do quantum witnesses offer more power than classical witnesses? Slightly more precisely, there are two natural ways to generalize NP from the classical setting to the quantum setting: QMA (for Quantum Merlin-Arthur) is the set of languages decidable by efficient quantum algorithms with quantum witnesses, whereas QCMA (for Quantum *Classical* Merlin-Arthur) is the set of languages decidable by efficient quantum algorithms with *classical* witnesses. A long-standing fundamental question, first raised by [2], is whether or not these two generalizations of NP are the same.

As an unconditional separation between QMA and QCMA is out of reach given the state of complexity theory, the community has focused on proving *oracle* separations. The first such separation [1] gives a *quantum* oracle; that is, an oracle that implements a unitary operation and which can be queried on quantum states. A major open question has been whether there is a *classical* oracle separation; that is, an oracle that implements a classical function, but is accessible in superposition. Classical oracle separations are considered more standard in the community.

An early candidate classical oracle separation was given by [9], but no proof was given. More recently, there have been several results making progress towards this goal by proving separations under different restrictions on how the oracle is accessed. [5] show a separation assuming the classical oracle is an "in-place permutation oracle", a non-standard modeling where the oracle irreversibly permutes the input state. [10] show a separation, assuming the witness is required to be independent of certain choices made in constructing the oracle. A very recent line of work has used quantum advantage relative to unstructured oracles [13] to separate QMA from QCMA: [8, 7] give a separation assuming the verifier can only make classical oracle queries, and more recently [3] give a separation which allows the verifier quantum queries, but assumes the adaptivity of the queries is sub-logarithmic. A standard classical oracle separation that makes no constraints on how the oracle is accessed or how the witness is created still remains open.


© Mark Zhandry;
licensed under Creative Commons License CC-BY 4.0



The central challenge in separating QMA from QCMA relative to a classical oracle seems to be the following. Consider a language in QMA \ QCMA, and consider measuring the QMA witness in the computational basis. The resulting classical string must *not* be accepted by the QMA verifier, since otherwise we would then have a classical witness for the language, putting it in QCMA. Therefore, in some sense, the QMA verifier needs to verify that the witness is in superposition. In order to allow for such verification using only a classical oracle, existing approaches require highly structured oracles. But then to actually prove the language is outside of QCMA, we need a quantum query complexity lower-bound, and unfortunately the techniques we have are often not amenable to highly structured oracles. Additionally, any witness would naturally be treated as a form of oracle-dependent advice about the oracle, and most quantum query complexity techniques are not very good at distinguishing between quantum advice and classical advice. In other words, if a typical technique succeeded at proving a language is outside of QCMA, then if it cannot distinguish quantum vs classical advice, it would likely also show that the language is outside QMA as well, thus failing to give a separation.

Our Work

We give a new approach for separating QMA from QCMA relative to a classical oracle. Like prior work, we are unable to prove our oracle separation unconditionally. However, our separation appears much less structured than the prior work (though this is an intuitive statement rather than a formal one), and appears much more amenable to existing quantum query complexity techniques. In particular, we prove under a natural conjecture about k-wise independent distributions that our separation indeed works. We believe our work adds to the evidence that QMA and QCMA are indeed distinct relative to classical oracles, and may offer a new path toward a proof.

1.1 Our Separating Oracles

Our basic idea is the following. An instance in our language will correspond to a subset $S \subseteq [N]$ where N is exponentially-sized. S will be chosen to only contain a negligible fraction of $[N]$, but still be super-polynomial sized. We will provide a classical oracle (accessible in superposition) that decides membership in S. We will often simply call this oracle S.

Next, we choose a random state $|\psi\rangle = \sum_{y \in S} \alpha_y |y\rangle$ with support on the set S. The state $|\psi\rangle$ will be our witness state for the QCMA instance. An incomplete verification for $|\psi\rangle$ proceeds by simply checking that $|\psi\rangle$ has support contained in S. Essentially, $|\psi\rangle$ is acting as a QMA witness that S is non-empty. But there is also a simple QCMA witness for this fact: any classical value $y \in S$.

We will instead attempt to turn $|\psi\rangle$ into a witness that S is *super-polynomial* sized. To do so, we will add a second oracle U which essentially attests to $|\psi\rangle$ being a superposition over super-polynomially-many points. Intuitively, we want to show that U can distinguish between $|\psi\rangle$ and any state whose support is only polynomial-sized.

To construct U, let $|\hat{\psi}\rangle$ denote the quantum Fourier transform (QFT) of $|\psi\rangle$. We observe that if $|\psi\rangle$ has support on a single point y, then $|\hat{\psi}\rangle$ will have uniform amplitude on all points in $[N]$ (though with complex phases on these points). On the other hand, for random $|\psi\rangle$ with support on the large set S, the amplitudes on different points will vary. Concretely, while the expected squared-amplitude on any point $z \in [N]$ is $1/N$, there is a reasonable chance that it could be, say, smaller than $1/2N$ or larger than $2/N$.

We will choose U to be a subset of $[N]$ consisting of points where $|\hat{\psi}\rangle$ has squared-amplitude somewhat larger than $1/N$. We can then have, say, the total squared-amplitude of $|\hat{\psi}\rangle$ on points in U be roughly $3/4$ while $|U|/N$ is only roughly $1/2$. In this case, the QFT of a classical string y will have squared-amplitude on U of only $1/2$. Thus, U enables distinguishing $|\psi\rangle$ from a classical input. We will therefore give out an oracle for deciding membership in U. The verifier will first confirm that $|\psi\rangle$ has support only on S using the oracle for S. Then it will compute $|\hat{\psi}\rangle$ via the QFT, and check that the support is in U. Overall the verifier accepts with probability $3/4$. Instances not in the language will consist of S, U pairs where S is very small but non-empty. We show that, for a certain way of choosing U, that there is no QMA witness in the case of such small S. Thus, our language is in QMA relative to the oracles for S, U.

1.2 QCMA hardness

We now need a way to argue that our language does not have QCMA witnesses. While we showed that a classical string $y \in S$ cannot serve as a witness, this alone does not preclude some more clever way of attesting to S being large. In particular, the witness w could contain several points in S. Worse, perhaps queries to U may reveal a significant amount of information about S, which may help deciding if S is large or small. We make progress toward showing QCMA hardness of our oracle problem, as we now describe.

Consider a hypothetical QCMA verifier V which is given a classical witness w and makes quantum queries to S, U, and accepts in the case S is large. We want to show that we can replace S with a small set S', and V will still accept with too-high a probability, meaning it incorrectly claims that (S', U) is in the language, despite S' being small.

Toward that end, we will choose S' to be all points in S that are also "heavy" among the queries V makes to S. That is, points $y \in S$ such that the query amplitude in V's quantum queries to S is above some inverse-polynomial threshold. As the total query amplitude of all points is just the number of queries of V and hence polynomial, the number of heavy y is polynomial. Hence, S' is small. We can also construct S' efficiently: for each heavy y, running V and measuring a random query will have an inverse polynomial chance of producing y. By repeating this process a polynomial number of times, we can collect all heavy queries.

But why should S and S' be indistinguishable to V? By standard quantum query analysis, if V can distinguish S from S', then it's queries must place significant amplitude on $S \setminus S'$. By measuring a random query, we therefore obtain with significant probability a $y \in S \setminus S'$. But since y is not heavy, repeating this process many times will produce many different y. This means that if V can distinguish S from S', it must actually be able to generate essentially arbitrarily large (polynomial) numbers of $y \in S$. Denote the number of y by L.

V that do not query U

Let us first assume that V makes no queries to U. In this case, we can argue that any distinguishing V actually violates known query complexity results for multiple Grover search. In particular, [6] show that an algorithm making polynomially-many queries to a random sparse S cannot produce L points in S except with probability bounded by 2^{-L} (see Lemma 8 for precise statement). Now, this result assumes no advice is provided about S, but the witness w counts as advice. Fortunately, we can handle the advice using the strong exponential lower bound provided by [6]. Consider running the process above with a *random* w' instead of w. In the event $w' = w$, the process will produce L points in S. Moreover, $w' = w$

with probability $2^{-|w|}$. By setting $L \gg |w|$ (recall that we can make L an arbitrarily large polynomial), we therefore obtain an algorithm with no advice which produces L points in S with probability $2^{-|w|} \gg 2^{-L}$, contradicting the hardness of multiple Grover search.

▶ Remark 1. The above strategy inherently uses the fact that w is classical. If V had a quantum witness/advice, running it even once and measuring a random query to find a $y \in S$ would potentially destroy the witness, meaning further runs of V are not guaranteed to produce *any* points in S. This is the key place in the proof where we distinguish between classical and quantum witnesses, hopefully indicating a promising route toward proving a separation between QMA and QCMA.

k-wise independent U

The above strategy does not work for handling queries to U. The problem is that U takes potentially N bits (which is exponential) to describe, meaning treating U as advice would require setting $L \gg N$, at which point the bounds from [6] do not apply.

We will for now assume that U is k-wise independent for a super-polynomial k, and return to justifying this assumption later. We will assume such k-wise independence holds even conditioned on S (but not conditioned on $|\psi\rangle$, whose correlation with U is crucial for the correctness of our QMA verifier). A result of [14] (formally described in Lemma 7) shows that a k-wise independent U is actually perfectly indistinguishable from a uniform U, for all quantum query algorithms making at most $k/2$ queries. Since k is super-polynomial, we thus obtain perfect indistinguishability against all polynomial-query algorithms, including our process above for generating points in S. Consequently, the process above succeeds in generating L points in S even if U is replaced by a uniformly random U independent of S. But such a uniform U can be simulated without knowledge of S at all, and hence the lower-bound of [6] actually applies to algorithms making queries to uniform U. Thus under the assumption that U is k-wise independent, we can justify QCMA hardness.

Our U are "close" to k-wise independent

We show that, by choosing U carefully in a probabilistic way, U is "close" to k-wise independent, even conditioned on S. By "close", we concretely mean that for every subset $T \subseteq [N]$ of size at most k, that $2^{-|T|} \leq \Pr[T \cap U = \emptyset] \leq (1+\epsilon) \times 2^{-|T|}$, for some very small ϵ which depends on $k, |S|, N$. Note that true k-wise independence is equivalent to $\Pr[T \cap U = \emptyset] = 2^{-|T|}$ for all T of size at most k.

Is this "close" enough?

Unfortunately, we do not know how to prove that U which are close to k-wise independent are sufficient to make our approach work. The issue is that [14] only applies to perfect k-wise independence, and there are counterexamples that show that the result does *not* hold when replaced with some approximate notions of k-wise independence.

The good news is that our notion of closeness is quite different from the usual notion of "biased" or "almost" k-wise independence used in the literature. Specifically, those notions allow for an additive error in any of the marginal probabilities, whereas we impose a strong *multiplicative* error bound. This gives us hope that our distribution of U, despite not being truly k-wise independent, may still be close enough to get a separation.

We make progress toward justifying this claim. We make a conjecture that any distribution over U which is "close" to k-wise independent (in our sense) can be turned into a distribution U' that is truly k-wise independent. Importantly, U and U' will agree on almost all points.

More precisely, for any z, the probability that U and U' differ on z is negligible. See Conjecture 10 for the formal statement of this conjecture. Observe that this conjecture is simply a statement about distributions, and has nothing on the surface to do with quantum query complexity.

Under this conjecture, we complete the full oracle separation between QMA and QCMA. Instead of giving out the oracle U, we simply give out the oracle U', and prove QCMA hardness following the above approach. Our statistical conjecture is then used to show that replacing U with U' does not break our QMA verifier. Concretely, by standard query complexity arguments, we show that if our verifier works on U, then it must also work (with negligibly larger error) on U'.

▶ Remark 2. Our statistical conjecture gives *one* way to prove an oracle separation between QMA and QCMA following our approach. Our conjecture could of course turn out to be false. Even in this case, our oracles still seem likely to give a separation, and there may be many other paths toward proving it. Perhaps if the general conjecture is false, our particular U can still be converted into U' as needed. Or maybe being "close" to k-wise independent is directly sufficient for a separation and can be proven via quantum query complexity arguments. Possibly there is a different distribution over $|\psi\rangle$ and associated U where U actually is perfectly k-wise independent. Thus, independent of our particular statistical conjecture, we believe our new approach at a separation gives a promising new approach toward separating QMA from QCMA relative to classical oracles.

2 Preliminaries and notation

A function $f(n) \leq n^{O(1)}$ is *polynomial* and $f(n) \geq n^{-O(1)}$ is inverse polynomial. Functions $f(n) \geq n^{\omega(1)}$ are superpolynomial and $f(n) \leq n^{-\omega(1)}$ are negligible.

Let $\mathsf{Bernoulli}_p$ denote the distribution over $\{0,1\}$ which outputs 1 with probability p. Let $\mathsf{Bernoulli}_p^N$ denote the distribution over $\{0,1\}^N$ consisting of N iid samples from $\mathsf{Bernoulli}_p$. We will associate $\{0,1\}^N$ with subsets of $[N]$, where $S \subseteq [N]$ is associated with the vector $\mathbf{v}$ such that $v_x = 1$ if $x \in S$. We will also associate $\{0,1\}^N$ (and hence also subsets of $[N]$) with functions from $[N]$ to $\{0,1\}$, where S is associated with its indicator function f, where $f(x)$ is 1 if and only if $x \in S$.

A joint distribution $X_1, \cdots X_n$ is k-wise independent if, for each subset T of size at most k, $(X_i)_{i\in T}$ are independent random variables. $X_1, \cdots, X_n$ is k-wise *uniform* independent if each $(X_i)_{i\in T}$ are independent uniform random variables over their respective domains.

Complex Normal Distribution

Let $\mathcal{N}^{\mathbb{C}}_{\mu,\sigma}$ denote complex normal distribution with width σ. The probability density function of this distribution is given by $\Pr[x \leftarrow \mathcal{N}^{\mathbb{C}}_{\mu,\sigma}] = \frac{1}{\pi\sigma^2} e^{-|x-\mu|^2/\sigma^2}$.

Two basic identities that will be useful when computing integrals involving complex normal distributions are the following. Let $\mathbf{v}$ is a vector of n complex numbers, $\mathbf{M}$ is a complex $n \times n$ matrix, and $\int_{\mathbb{C}} d\mathbf{v}$ means to integrate over all complex vectors $\mathbf{v}$. Then

$$\int_{\mathbb{C}} e^{-\mathbf{v}^\dagger \mathbf{M} \mathbf{v}} d\mathbf{v} = \frac{\pi}{\det(\mathbf{M})} \qquad \int_{\mathbb{C}} |v_1|^2 |v_2|^2 e^{-\mathbf{v}^\dagger \mathbf{M} \mathbf{v}} d\mathbf{v} = \frac{\pi^2 \mathsf{Tr}(\mathbf{M})^2}{4 \det(\mathbf{M})^3} \text{ for } n = 2$$

Quantum Computation

We assume the reader is familiar with the basics of quantum computation and quantum query models. Recall that in the standard model for making quantum queries to a classical oracle $\mathcal{O}$, the algorithm submits a state $\sum_{x,y,z} \alpha_{x,y,z} |x,y,z\rangle$, and receives back $\sum_{x,y,z} \alpha_{x,y,z} |x, y \oplus$

$\mathcal{O}(x), z\rangle$. We will make use of the quantum Fourier tranform, denoted QFT_N, defined on computational basis states as $|y\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{z \in \mathbb{Z}_N} e^{i2\pi yz/N} |z\rangle$ for $y \in \mathbb{Z}_N$. We will usually drop the subscript N as it will be clear from context. For a quantum state $|\psi\rangle$, we will typically let $|\hat{\psi}\rangle$ be shorthand for $\mathsf{QFT}|\psi\rangle$.

2.1 Defining QMA and QCMA relative to Oracles

We can consider two types of oracle versions of complexity classes, and in particular QMA/QCMA. The first, and most common, is to specify an infinite oracle $\mathcal{O} : \{0,1\}^* \to \{0,1\}$, and define the classes QMA/QCMA relative to $\mathcal{O}$. The second version, which is typically easier to work with, is to consider a variant of QMA/QCMA where the instance itself is an exponentially-sized oracle $\mathcal{X} : \{0,1\}^n \to \{0,1\}$. Fortunately, we show that a QMA/QCMA separation for one variant immediately gives such a separation for the other variant. Thus, it suffices to prove a separation for whichever variant is most convenient.

▶ **Definition 3** (Oracle-Aided QMA/QCMA)**.** *For a function $\mathcal{O} : \{0,1\}^* \to \{0,1\}$, an oracle decision problem is a subset $\mathsf{L}^{\mathcal{O}} \subseteq \{0,1\}^*$ which depends on $\mathcal{O}$. We say that $\mathsf{L}^{\mathcal{O}}$ is in $\mathsf{QMA}^{\mathcal{O}}$ if there exists a polynomial-time oracle-aided quantum algorithm $V^{\mathcal{O}}$ and polynomial p such that:*

- *For every $x \in \mathsf{L}^{\mathcal{O}}$ of length n, there exists a state $|\psi\rangle$ on $p(n)$ qubits such that $\Pr[V^{\mathcal{O}}(x, |\psi\rangle) = 1] \geq 2/3$. In this case, we say that $V^{\mathcal{O}}$* accepts *x.*
- *For every $x \notin \mathsf{L}^{\mathcal{O}}$ of length n, and for any state $|\psi\rangle$ on $p(n)$ qubits, $\Pr[V^{\mathcal{O}}(x, |\psi\rangle) = 1] \leq 1/3$. In this case, we say that $V^{\mathcal{O}}$* reject *x.*

$\mathsf{QCMA}^{\mathcal{O}}$ is defined analogously, except that the states $|\psi\rangle$ are required to be classical.

▶ **Definition 4** (Oracle-Input QMA/QCMA)**.** *Let $\mathcal{U} = \{\mathcal{U}_n\}_{n \in \mathbb{Z}^+}$ where $\mathcal{U}_n$ are sets of strings $\mathcal{X}$ of length $2^{n^{\Theta(1)}}$, interpreted as functions from $\mathcal{X} : [n^{\Theta(1)}] \to \{0,1\}$. An oracle-input promise language is a subset $\mathsf{OI\text{-}L} \subseteq \mathcal{U}$. An oracle-input promise language $\mathsf{OI\text{-}L} \subseteq \mathcal{U}$ is in $\mathsf{OI\text{-}QMA}$ if there exists a polynomial-time oracle-aided quantum algorithm V and polynomial p such that:*

- *For every $\mathcal{X} \in \mathsf{OI\text{-}L} \cap \mathcal{U}_n$, there exists a $|\psi\rangle$ on $p(n)$ qubits such that $\Pr[V^{\mathcal{X}}(|\psi\rangle) = 1] \geq 2/3$.*
- *For every $\mathcal{X} \in \mathcal{U}_n \setminus \mathsf{OI\text{-}L}$, and for any state $|\psi\rangle$ on $p(n)$ qubits, $\Pr[V^{\mathcal{X}}(|\psi\rangle) = 1] \leq 1/3$.*

$\mathsf{OI\text{-}QCMA}$ is defined analogously, except that the states $|\psi\rangle$ are required to be classical.

Note that the constants $1/3, 2/3$ in Definitions 3 and 4 are arbitrary, and can be replaced with a, b for any a, b such that $a \geq 2^{-\mathsf{poly}(n)}, b \leq 1 - 2^{-\mathsf{poly}(n)}$, and $b - a \geq 1/\mathsf{poly}(n)$.

Given any countable collection of oracles $\mathcal{O}_1, \mathcal{O}_2, \cdots$ which may have finite or infinite domains, we can construct a single oracle $\mathcal{O} : \{0,1\}^* \to \{0,1\}$ by setting $\mathcal{O}(j, x) = \mathcal{O}_i(x)$, where (j, x) is some encoding of pairs $(j, x) \in \{0,1\}^* \times \{0,1\}^*$ into strings in $\{0,1\}^*$. We can likewise convert $\mathcal{O}$ back into $\mathcal{O}_1, \mathcal{O}_2, \cdots$. Therefore, we will take the definitions of $\mathsf{QMA}, \mathsf{QCMA}, \mathsf{OI\text{-}QMA}, \mathsf{OI\text{-}QCMA}$ to allow for verifiers making queries to countable collections of oracles.

This next theorem is proved in Section 4 following a standard diagonalization argument, and shows that is suffices to give a separation for either variant of $\mathsf{QMA}, \mathsf{QCMA}$.

▶ **Theorem 5.** *There exists a classical oracle $\mathcal{O}$ such that $\mathsf{QCMA}^{\mathcal{O}} \neq \mathsf{QMA}^{\mathcal{O}}$ if and only if $\mathsf{OI\text{-}QCMA} \neq \mathsf{OI\text{-}QMA}$.*

The style of oracle separation between QMA and QCMA given in [1] follows that of Definition 3, except that they use a quantum oracle instead of a classical oracle. They do not define the oracle-input versions of $\mathsf{OI\text{-}QMA}$ and $\mathsf{OI\text{-}QCMA}$ or a general result like Theorem 5, but their

proof implicitly uses similar concepts. In particular, they first define a universe of quantum oracles: namely, those that are either the identity or reflect around a Haar random state. They then show that the two cases can be efficiently distinguished via a quantum witness, but not a classical witness. This is effectively showing a separation between OI-QMA and OI-QCMA, except using quantum oracles instead of classical oracles. They then extend this to give a quantum oracle $\mathcal{Q}$ separating $\mathsf{QMA}^{\mathcal{Q}}$ from $\mathsf{QCMA}^{\mathcal{Q}}$. This can be seen as roughly corresponding to a transformation like what we prove in Theorem 5.

One slight non-triviality in generalizing their techniques to give a general equivalence is that a separation between OI-QMA and OI-QCMA only needs to show that, for any potential QCMA verifier, there is *some* instance that the verifier answers incorrectly. This is indeed how our separation is constructed. In contrast, [1] show that almost all instances will fool any QCMA verifier. This stronger separation is crucially used when constructing the oracle $\mathcal{Q}$ separating $\mathsf{QMA}^{\mathcal{Q}}$ from $\mathsf{QCMA}^{\mathcal{Q}}$, as they can basically choose $\mathcal{Q}$ to be random from the appropriate universe of oracles. In order to get a general equivalence for arbitrary separations, including those like ours where the instance depends on the verifier, we have to work a bit harder and incorporate a diagonalization argument. Fortunately, this is standard.

2.2 Some Useful Quantum Lemmas

For an oracle algorithm A making queries to an oracle O and an oracle input r, let the state $\sum_{x,y,z} \alpha^{(i)}_{x,y,z}|x,y,z\rangle$ denote the ith oracle query, and let $M_x(i) = \sum_{y,z} |\alpha^{(i)}_{x,y,z}|^2$, which we will call the **query mass** of x in the i-th query. Let $M_x = \sum_i M_x(i)$, the total query mass of x over all q queries, and for a subset V let $M_V = \sum_{x\in V} M_x$ be the total query mass of points in V.

▶ **Lemma 6** ([4] Theorem 3.1+3.3). *Let A be a quantum algorithm running making q queries to an oracle O. Let $\epsilon > 0$ and let V be a set of inputs. If we modify O into an oracle O' which is identical to O except possibly on inputs in V, then $|\Pr[A^O() = 1] - \Pr[A^{O'}() = 1]| \leq 4\sqrt{qM_V}$.*

▶ **Lemma 7** ([14] Theorem 3.1). *Let A be a quantum algorithm making q queries to an oracle O. For any output z, the probability A outputs z when O is $2q$-wise independent is identical to the probability A outputs z when O is uniformly random.*

▶ **Lemma 8** ([6] Theorem 5.5). *Let $p \in [0,1]$. There is a constant $C \leq 48e$ such that the following is true. The success probability of finding K marked items in a random function $S : [N] \to \{0,1\}$ where $S(x) = 1$ with probability p for each $x \in [N]$ is at most $(Cp(Q/K)^2)^K$ for any algorithm making $Q \geq K$ quantum queries to S.*

3 Our Oracle Separation

Here, we give our conjectured oracle separation between QMA and QCMA. Following Theorem 5, it suffices to focus on the oracle-input variants of QMA and QCMA. We first define our new statistical conjecture. Then we will define a certain "nice" type of distribution, which we call Fourier Independent (FI). We show that our statistical conjecture leads to the existence of such a FI distribution. Finally, we show that an FI distribution gives a separation between OI-QMA and OI-QCMA, and hence an oracle $\mathcal{O}$ such that $\mathsf{QCMA}^{\mathcal{O}} \neq \mathsf{QMA}^{\mathcal{O}}$ (via Theorem 5).

3.1 Our Statistical Conjecture

▶ **Definition 9** (Substitution Distance). *Consider two distributions X_0, X_1 over Σ^ℓ for some alphabet Σ. The* substitution distance *between X_0 and X_1, denoted $\|X_0 - X_1\|_{\mathsf{sub}}$, is the minimum value of $\epsilon \geq 0$ such that there is a joint distribution (Z_0, Z_1) over $(\Sigma^\ell)^2$ where:*

- *The marginal distribution Z_0 is equal to X_0 and the marginal distribution Z_1 is equal to X_1.*
- *For each $i \in [\ell]$, $\Pr[Z_{0,i} \neq Z_{1,i}] \leq \epsilon$, where $Z_{b,i}$ means the ith entry of Z_b.*

Intuitively, a small $\|X_0 - X_1\|_{\mathsf{sub}}$ means that, by changing a few positions X_0 – and each position with tiny probability – we can turn X_0 into X_1.

▶ **Conjecture 10.** *There exist functions $r(N), k(N), \zeta(N), \eta(N)$ subject to the constraints $k(N) \geq (\log N)^{\omega(1)}$, $\eta(N) \leq (\log N)^{-\omega(1)}$, and $N\zeta(N)^3/r(N)^6 \geq (\log N)^{\omega(1)}$ such that the following is true. Suppose X_0 is a distribution over $\{0,1\}^N$ such that for all sets $T \subseteq [N]$ of size of size at most $r(N)$, $2^{-|T|} \leq \Pr_{\mathbf{x} \leftarrow X_0}[x_i = 0 \forall i \in T] \leq (1 + \zeta(N)) \times 2^{-|T|}$. Then there exists a $k(N)$-wise uniform independent distribution X_1 such that $\|X_0 - X_1\|_{\mathsf{sub}} \leq \eta(N)$.*

Think of $\log N$ as the instance size, so that N is exponential. Conjecture 10 starts with a distribution X_0 that is in some sense very close to r-wise uniform independent, and concludes that X_0 must be negligibly-close in substitution error to an actual k-wise uniform independent distribution, for k that may be different than r but is still super-polynomial. Note that without the constraint $N\zeta(N)^3/r(N)^6 \geq (\log N)^{\omega(1)}$, the conjecture is trivially true by setting $\zeta = 0$ and $X_0 = X_1$. The exact constraint we stipulate makes the conjecture non-trivial, and arises for technical reasons in our separation.

3.2 Fourier Independent Distributions

▶ **Definition 11.** *Let $N \in \mathbb{Z}^+$ and $S \subseteq [N]$. We say that a distribution $\mathcal{D}_S$ over pairs $(|\psi\rangle, U)$ is (k, δ, γ)-Fourier-Independent (FI) if the following hold:*

- *$|\psi\rangle$ is a normalized superposition $\sum_{y \in S} \alpha_y |y\rangle$ over elements $y \in S$.*
- *$U \subseteq [N]$, and the marginal distribution of U is k-wise uniform independent.*
- *Let $|\hat{\psi}\rangle = \mathsf{QFT}|\psi\rangle$ be the quantum Fourier transform of $|\psi\rangle$. Then except with probability δ over the choice of $|\psi\rangle$, $\langle\hat{\psi}|\Pi_U|\hat{\psi}\rangle \geq \frac{1}{2} + \gamma$, where Π_U is the projection operator $\sum_{z \in U} |z\rangle\langle z|$.*

We want k to be large (say super-polynomial in $\log N$), δ to be small (say negligible in $\log N$), and γ to be not-too-small (non-negligible in $\log N$). Then a Fourier Independent distribution is one where (1) $|\psi\rangle$ has support on S, (2) U looks like a random set to query-bounded quantum algorithms (by Lemma 7), but (3) $|\hat{\psi}\rangle$ is biased toward elements in U.

We now show the existence of a FI distribution for certain k, δ, γ, assuming Conjecture 10.

▶ **Theorem 12.** *Suppose Conjecture 10 holds. Then there exists functions $p, \epsilon, \delta, \gamma : \mathbb{Z}^+ \to [0,1]$ and functions $N, k : \mathbb{Z}^+ \to \mathbb{Z}^+$ such that (1) $p(n), \epsilon(n), \delta(n) \leq n^{-\omega(1)}$, (2) $N(n) \leq 2^{n^{O(1)}}$ (3) $k(n) \geq n^{\omega(1)}$, (4) $\gamma \geq n^{-O(1)}$, and (5) except with probability $\epsilon(n)$ over the choice of $S \leftarrow \mathsf{Bernoulli}_{p(n)}^{N(n)}$, there exists a distribution $\mathcal{D}_S$ that is (k, δ, γ)-Fourier-Independent.*

Proof. Let $N, \ell \in \mathbb{Z}^+$ be positive integers. For a subset $S \subseteq \mathbb{Z}_N$ of size ℓ, let $\mathcal{D}'_S$ be the distribution over pairs $(|\psi\rangle, U')$ where $|\psi\rangle$ is a quantum state and $U' \subseteq \mathbb{Z}_N$, obtained as follows:

- For each $y \in S$, sample a complex number $\alpha_y \leftarrow \mathcal{N}^{\mathbb{C}}(0, \sigma)$ where $\sigma = 1/\sqrt{\ell}$. Let $|\psi\rangle = \sum_{y \in S} \alpha_y |y\rangle$
- For each $z \in \mathbb{Z}_N$, let $\beta_z = \sum_{y \in S} e^{i2\pi yz/N} \alpha_y$.
- For each $z \in \mathbb{Z}_N$, place z into the output set U' with independent probability $1 - e^{-|\beta_z|^2}$.

Intuition

By choosing $\sigma = 1/\sqrt{\ell}$, we will show that $|\psi\rangle$ is approximately normalized, so we can think of $|\psi\rangle$ as being essentially a random state with support on S. Let $|\hat{\psi}\rangle = \mathsf{QFT}|\psi\rangle$ be the QFT applied to $|\psi\rangle$. Then $|\hat{\psi}\rangle = \sum_{z\in\mathbb{Z}_N}(\beta_z/\sqrt{N})|z\rangle$. Thus, the set U' will be biased toward containing the points z where β_z is large. We will also show that U' is close to k-wise independent in substitution distance, for an appropriate k. Via Conjecture 10, this allows us to replace U' with an appropriate U, giving a distribution $\mathcal{D}_S$ that is truly Fourier Independent.

We state some useful lemmas about $\mathcal{D}'_S$; the proofs will be given in Section 4. Define $\tau_T^S := \Pr_{U'\leftarrow\mathcal{D}_S}[T\cap U' = \emptyset]$, or equivalently, the probability $U'(z) = 0$ for all $z\in T$, once S is chosen.

▶ **Lemma 13.** *For any subsets S, T, $1 + |T| \le (\tau_T^S)^{-1} \le 2^{|T|}$*

Lemma 13 holds for any S, T. In particular, for $|T| = 1$, we have that $\tau_T^S = 1/2$. This means each z is placed in U with probability $1/2$; by linearity of expectation, $\mathbb{E}[|U'|] = N/2$. We can also get a tighter lower-bound for larger T with high probability over the choice of S:

▶ **Lemma 14.** *For any $\epsilon > 0$, except with probability at most $2N^2e^{-\epsilon^2\ell/2}$ over the choice of random subset S of size ℓ, for all subsets T, $(\tau_T^S)^{-1} \ge 2^{|T|}(1 - |T|^2\epsilon)$. In this event, as long as $|T|^2\epsilon \le 1/2$, we can bound $\tau_T^S \le 2^{-|T|}(1 + 2|T|^2\epsilon)$.*

Now we bound $\| |\psi\rangle \|^2$, showing that $|\psi\rangle$ is approximately normalized.

▶ **Lemma 15.** *Fix any set S of size ℓ and $\epsilon \in (0,1)$. Then except with probability $2e^{-\epsilon^2\ell/8}$, $\| |\psi\rangle \|^2 \in [1-\epsilon, 1+\epsilon]$, where $|\psi\rangle$ is generated as in $\mathcal{D}'_S$.*

Now, we bound the probability that (the normalization of) $|\hat{\psi}\rangle$ is accepted by U'. Note that $| |\psi\rangle |^2 = | |\hat{\psi}\rangle |^2$.

▶ **Lemma 16.** *Except with probability at most $3(N^{-1} + \ell^2N^{-2})\epsilon^{-2} + 4N^2e^{-\ell\epsilon^2/32}$ over the choice of S, U', we have $\left|\frac{\langle\hat{\psi}|\Pi_{U'}|\hat{\psi}\rangle}{\| |\psi\rangle \|^2} - 3/4\right| \le \epsilon$. Recall $\Pi_{U'}$ is the projection operator $\sum_{z\in U'}|z\rangle\langle z|$.*

We now return to the proof of Theorem 12. $\mathcal{D}'_S$ is almost Fourier Independent, except that the distributions over U' are not k-wise uniform independent. However, Lemmas 13 and 14 show, in some sense, that U' is very close to being k-wide uniform independent, for somewhat large k. We will then invoke the assumed Conjecture 10 to replace U' with a distribution over U that is actually k-wise uniform independent, for a sufficiently large k, and is close in substitution error to U'. We then show that the small substitution error between U and U' implies that the statements of Lemma 16 still approximately holds, even for U.

In more detail, let X_0 be the distribution over U' stemming from $\mathcal{D}'_S$. Let $r(N)$, $k(N)$, $\zeta(N)$, $\eta(N)$ be the functions guaranteed by Conjecture 10. Define $N = \Theta(2^n)$ and let $p = r(N)^4\zeta(N)^{-2}N^{-1}\log^2 N$. By the conditions Conjecture 10 places on $r(N), k(N), \zeta(N), \eta(N)$, we therefore have that $p \le n^{-\omega(1)}\log^2 N \le n^{-\omega(1)}$. By standard concentration inequalities, the size ℓ of S sampled from $\mathsf{Bernoulli}_p^N$ is very close to $pN = r(N)^4\zeta(N)^{-2}\log^2 N$, except with negligible probability. Then set $\epsilon = \zeta(N)/r(N)^2$, which by the conditions of Conjecture 10 gives that $N^{-1}\epsilon^{-2}$, $\ell^2N^{-2}\epsilon^{-2}$, and $N^2e^{-\ell\epsilon^2/32}$ are all negligible. By Lemma 13 and by invoking Lemma 14 with $\epsilon = \zeta(N)/r(N)^2$, we have except with negligible probability over the choice of S, that $2^{-|T|} \le \tau_S^T \le 2^{-|T|}(1+\zeta)$ for all sets T of size at most $r(N)$. Conjecture 10 then implies for "good" S where the bounds on τ_S^T hold, there is a distribution X_1 that is $k(N)$-wise uniform independent and such that $\|X_0 - X_1\|_{\mathsf{sub}} \le \eta(N)$, with $k(N), 1/\eta(N) \ge (\log N)^{\omega(1)} = n^{\omega(1)}$.

Let S be a good set. Now consider the following distribution $\mathcal{D}_S$. It first uses the fact that $\|X_0 - X_1\|_{\mathsf{sub}} \leq n^{-\omega(1)}$ to derive a joint distribution (Z_0, Z_1) with the marginal Z_0 being equivalent to X_0 and Z_1 being k-wise uniform independent for $k \geq n^{\omega(1)}$. Then we sample $(|\psi\rangle, U') \leftarrow \mathcal{D}'_S$ and sample U from the distribution Z_1 conditioned on $Z_0 = U'$. Let $|\psi'\rangle = |\psi\rangle / \| |\psi\rangle \|$. Output $(|\psi'\rangle, U)$. We then have that U is distributed according to Z_1, and is therefore k-wise uniform independent. We also have by Lemma 16 that except with negligible probability, $\langle\hat{\psi}'|\Pi_{U'}|\hat{\psi}'\rangle \geq 2/3$. In order to justify Fourier Independence, we just need to show that, e.g. $\langle\hat{\psi}'|\Pi_U|\hat{\psi}'\rangle \geq 7/12$.

Toward that end, write $|\hat{\psi}'\rangle = \frac{1}{\sqrt{N}} \sum_{z \in \mathbb{Z}_N} \beta'_z |z\rangle$ where $\sum_z |\beta'_z|^2 = 1$. Then deinfe

$$E := \langle\hat{\psi}'|\Pi_{U'}|\hat{\psi}'\rangle - \hat{\psi}'|\Pi_U|\hat{\psi}'\rangle = \sum_{z \in U'} |\beta'_z|^2 - \sum_{z \in U} |\beta'_z|^2 \leq \sum_{z \in U' \setminus U} |\beta'_z|^2$$

Now let ξ_z denote the variable that is 1 if $z \notin U' \setminus U$ and is 0 otherwise. Then $E \leq \sum_z \xi_z |\beta'_z|^2$. Each ξ_z is a 0/1 random variable with negligible expectation. Therefore, since $\sum_z |\beta'_z|^2 = 1$, we have that $\mathbb{E}[E]$ is negligible. Then by Markov's inequality, we have that $\Pr[E \geq \sqrt{\mathbb{E}[E]}] \leq \sqrt{\mathbb{E}[E]}$. Therefore, since $\mathbb{E}[E]$ and hence $\sqrt{\mathbb{E}[E]}$ are negligible, we have that E is negligible except with negligible probability. This means in particular that $\langle\hat{\psi}'|\Pi_U|\hat{\psi}'\rangle \geq 7/12$, except with negligible probability. Thus $\mathcal{D}_S$ is (k, δ, γ)-Fourier-Independent, thereby proving Theorem 12. ◀

3.3 From FI Distributions to An Oracle Separation

We now show that FI distributions as guaranteed by Theorem 12 give a separation between OI-QCMA and OI-QMA.

▶ **Theorem 17.** *Suppose Conjecture 10 holds. Then* OI-QCMA $\neq$ OI-QMA.

Proof. We first invoke Theorem 12 to obtain distributions $\mathcal{D}_S$ satisfying properties (1) through (5) in the statement of Theorem 12. We now use this to construct our separation. We will invoke the definition of OI-QMA and OI-QCMA with $a = 1/2 + \gamma(n)/2$ and $b = 1/2 + \gamma(n)$. Since $b - a = \gamma(n)/2$ is inverse polynomial, this is equivalent to the standard definition of OI-QMA and OI-QCMA. Thus, a valid verifier will accept YES instances of size n (given a correct witness) with probability at least $1/2+\gamma(n)$, and accept NO instances with probability at most $1/2 + \gamma(n)/2$.

An instance will be a pair of oracles (S, U) where $S, U \subseteq [N(n)]$. Our verifier $V^{S,U}(|\psi\rangle)$ will do the following: V will make a query to S on the witness state $|\psi\rangle$ and measuring the output. If it accepts, then V will apply the QFT to the witness state (which may have changed due to the measurement), and make a query to U, measuring the output. If both queries accept, then V will output 1; if either measurement rejects, then V will output 0.

We define the universe $\mathcal{U}$ as the set of pairs (S, U) for which either (1) there exists a state $|\psi\rangle$ such that $\Pr[V^{S,U}(|\psi\rangle) = 1] \geq 1/2 + \gamma(n)$, or (2) for all states $|\psi\rangle$, $\Pr[V^{S,U}(|\psi\rangle) = 1] < 1/2 + \gamma(n)/2$. Then OI-L $\subseteq \mathcal{U}$ is the set such that $\Pr[V^{S,U}(|\psi\rangle) = 1] \geq 1/2 + \gamma(n)$. By definition, OI-L $\in$ OI-QMA.

We now show that OI-L $\notin$ OI-QCMA. We can always assume without loss of generality that there is, say, a fixed quadratic running time t such that the running time of any OI-QCMA verifier is bounded by $t(|x| + |w|)$ where $|x|$ is the instance length and $|w|$ is the witness length. This is accomplished by padding the witness length with 0's that just get ignored by the verifier.

Now let $q = q(n)$ be a sufficiently small super-polynomial function, which we will take as an upper bound on witness length. Let $Q(n) = q(n)^2$, which we take to be an upper bound on the number of queries when the witness length is at most $q(n)$. Let $v = v(n)$ be another super-polynomial. We will need the following constraints on q, Q, v:

$$Q \ll \gamma p^{-1/2} \qquad v \gg qQ^4\gamma^{-4}$$
$$v \ll \gamma N^{1/12} \qquad Qk/v \leq n^{-\omega(1)}$$

These can be satisfied for a sufficiently-small super-polynomial q, v and for sufficiently large n.

Suppose toward contradiction that there is a $\mathsf{OI\text{-}QCMA}$ verifier V_*. Sample $S \leftarrow \mathsf{Bernoulli}_{p(n)}^{N(n)}$ and $(|\psi\rangle, U) \leftarrow \mathcal{D}_S$, where $\mathcal{D}_S$ is $(k(n), \delta(n), \gamma(n))$-Fourier-Independent. Then we have that with overwhelming probability over the choice of $S, U, |\psi\rangle$, $\Pr[V^{S,U}(|\psi\rangle) = 1] \geq 1/2 + \gamma(n)$. Hence $(S, U) \in \mathsf{OI\text{-}L}$. Therefore, under the assumption that V_* is a verifier for $\mathsf{OI\text{-}L}$, V_* must accept (S, U), meaning there is a classical witness w such that $\Pr[V_*^{S,U}(w) = 1] \geq 1/2 + \gamma(n)$.

We will now construct a different instance S', U, where S' is constructed from the following algorithm $\mathsf{GenSmallSet}^{S,U}(w)$: initialize $S' = \{\}$, and then repeat the following loop for $i = 1, ..., v$:

- Run $V_*^{S,U}(w)$ until a randomly chosen query to S, and measure the query, obtaining a string $y_i \in [N]$.
- If $y_i \in S \setminus S'$, add y_i to S'.

The following is proved in Section 4:

▶ **Lemma 18.** *If U is sampled from a k'-wise uniform independent function and S' is potentially correlated to U but has size at most v, then except with probability $2N^2 \left(\frac{\sqrt{ek'}}{\epsilon\sqrt{N}}\right)^{k'}$ over the choice of U, S', it holds that, for any normalized state $|\phi\rangle$ with support on S', $\langle\hat{\phi}|\Pi_U|\hat{\phi}\rangle \leq 1/2 + v\epsilon$, where $|\hat{\phi}\rangle = \mathsf{QFT}|\phi\rangle$.*

▶ **Corollary 19.** *Except with negligible probability over the choice of S, U, S', $(S', U) \in \mathcal{U} \setminus \mathsf{OI\text{-}L}$.*

Proof. Set $\epsilon = \gamma/3v$, and set $G := 2N^2 \left(\frac{3v\sqrt{ek'}}{\gamma\sqrt{N}}\right)^{k'}$ for a yet-unspecified k'. By Lemma 18, except with probability at most G, $\Pr[V^{S',U}(|\phi\rangle) = 1] \leq 1/2 + \gamma/3 < 1/2 + \gamma/2$ for any state $|\phi\rangle$. In this case, $(S', U') \in \mathcal{U} \setminus \mathsf{OI\text{-}L}$. This event holds regardless of what S' is, just using the fact that it consists of at most v elements. Setting $k' = 5$ and using that $v \leq \gamma N^{1/12}$ we have that $G = O(N^{1/12})$, which is negligible. ◀

We now show, however, that V^* fails to reject S', U:

▶ **Lemma 20.** *Except with negligible probability over the choice of S, U, w, S', $\Pr[V_*^{S',U}(w) = 1] > 1/2 + \gamma(n)/2$*

Proof. Observe that the process $\mathsf{GenSmallSet}^{S,U}(w)$ for constructing S' always generates $S' \subseteq S$. Now consider running $S' \leftarrow \mathsf{GenSmallSet}^{S,U'}(w')$, where w' is a random string and U' is a random boolean function, with both w', U' independent of S. This is an algorithm which makes vQ queries to S (and also to U', but U' us simulatable on its own since it is independent of S). S in turn sampled from $\mathsf{Bernoulli}_p^N$. Finally, the algorithm outputs some subset S' of S. By Lemma 8, there is a universal constant C such that

$$\Pr[|S'| \geq K : w', U' \text{ are uniform and independent of } S] \leq (Cp(vQ/K)^2)^K.$$

Now consider running $S' \leftarrow \mathsf{GenSmallSet}^{S,U}(w')$, replacing U' with U. Recall that U is k-wise independent even conditioned on S, for $k \geq 2vQ$ where vQ is the number of queries made by $\mathsf{GenSmallSet}$. By Lemma 7, the output distribution of $\mathsf{GenSmallSet}^{S,U}(w')$ is identical to that of $\mathsf{GenSmallSet}^{S,U'}(w')$. Thus, we still have

$$\Pr[|S'| \geq K : U' = U, \text{ while } w' \text{ is uniform and independent of } S] \leq (Cp(vQ/K)^2)^K.$$

Finally, consider running $S' \leftarrow \mathsf{GenSmallSet}^{S,U}(w)$. Since $\Pr[w' = w] = 2^{-q}$, this means that with probability 2^{-q}, $\mathsf{GenSmallSet}^{S,U}(w')$ is actually running $\mathsf{GenSmallSet}^{S,U}(w)$. Therefore,

$$\Pr[|S'| \geq K : U' = U, w' = w] \leq (Cp(vQ/K)^2)^K \times 2^q.$$

We now set $K = vQ\sqrt{2Cp} + q + \log_2(v)$. Then we have that $\Pr[|S'| \geq K : U' = U, w' = w] \leq 1/v$. Then in particular since $|S'| \leq v$ always, we have that $\mathbb{E}[|S'|] \leq (K-1)\Pr[|S'| < K] + v\Pr[|S'| \geq K] \leq (K-1) + 1 \leq K$.

Let e_j be the probability that $\mathsf{GenSmallSet}$ adds an element y_i to S' in the ith iteration. Then $\sum_{j=1}^{v} e_j = \mathbb{E}[|S'|] \leq K$. We also have that the e_j are monotonically decreasing since the y_i are identically distributed and thus the probability mass outside of the growing S' can only shrink. Thus $e_v \leq K/v$.

For an input $y \in S$, let M_y denote the total magnitude squared of y in all queries made by $V_*^{S,U}(w)$ to oracle S. Then measuring a random choice of the $\leq Q$ queries by V_*, we will obtain y with probability at least M_y/Q. Since $e_v \leq K/v$, this means that by the end of the loop, the expected total query weight of all points in S but not in S' is at most QK/v. By Markov's inequality, we therefore have that the query weight of points in $S \setminus S'$ is at most $\sqrt{QK/v}$, except with probability at most $\sqrt{QK/v}$. Recall that $\sqrt{QK/v}$ is negligible. Therefore, since this probability is negligible, we will therefore assume the total query weight of points in $S \setminus S'$ is at most $\sqrt{QK/v}$.

Lemma 6 shows that the difference in acceptance probability between $V_*^{S,U}(w)$ and $V_*^{S',U}(w))$ is at most $4\sqrt{Q \times \sqrt{QK/v}} = 4\sqrt[4]{Q^3K/v}$. Observe that $Q^3K/v = Q^4\sqrt{2Cp} + Q^3(q + \log_2(v))/v$. Using that $Q \ll \gamma p^{-1/2}$ and $v \gg qQ^4/\gamma^4$, we therefore have that $Q^3K/v \leq \gamma^4/(12)^4$. Hence the difference in acceptance probability is at most $\gamma/3$, and hence $\Pr[V_*^{S',U}(w)) = 1] > 1/2 + \gamma/2$. ◀

Corollary 19 and Lemma 20 shows that V_* fails to decide $\mathsf{OI\text{-}L}$, contradicting it being a QCMA verifier. Hence, $\mathsf{OI\text{-}L} \notin \mathsf{QCMA}$. This completes the proof of Theorem 17 ◀

4 Missing Proofs

We now prove Lemmas 13, 14, 15, and 16 as well as Theorem 5. To do so, we introduce some extra notation. For a subset $S \subseteq \mathbb{Z}_N$, let $\mathbf{M}^S$ be the $N \times N$ matrix defined as $\mathsf{QFT}^\dagger \circ \Pi_S \circ \mathsf{QFT}$. Define $\tau_T^S := \Pr_{U' \leftarrow \mathcal{D}'_S}[T \cap U = \emptyset]$, or equivalently, the probability $U'(z) = 0$ for all $z \in T$, once S is chosen. Let $\mathcal{M}_T^S$ be the $|T| \times |T|$ sub-matrix of $\mathbf{M}^S$ consisting of rows and columns columns whose indices are in T. The following lemmas will be useful.

▶ **Lemma 21.** *For any $\epsilon > 0$, except with probability at most $2N^2 e^{-\epsilon^2 N^2/8\ell}$ over the choice of random S of size ℓ, $\max_{z \neq z'} |\mathbf{M}^S_{z,z'}| \leq \epsilon$*

Proof. Fix any $z \neq z'$, and consider the random variable $M^S_{z,z'}$ where S is a random set of size ℓ. We write:

$$M^S_{z,z'} = \frac{1}{N}\sum_{y\in S} e^{i2\pi(z-z')y/N} = \frac{1}{N}\sum_{j=1}^{\ell} e^{i2\pi(z-z')y_j/N}$$

where $y_1 \cdots, y_\ell$ range over the elements of S, and are therefore random distinct values in $\mathbb{Z}_N$.

We first look at the real part of $\mathbf{M}^S_{z,z'}$, namely $\frac{1}{N}\sum_{i=1}^{\ell} Y_i$ where $Y_i = \cos(2\pi(z-z')y_i/N)$. For the moment, imagine the y_i being uniform without the distinctness requirement, in which case since $z - z' \neq 0$ the Y_i all become independent random variables bounded to the range $[-1, 1]$. Moreover, the means are zero since $z - z' \neq 0 \bmod N$. Then by Hoeffding's inequality, we have

$$\Pr\left[\left|\sum_{i=1}^{\ell} Y_i\right| \geq t\right] \leq 2e^{-t^2/4\ell}$$

We then observe that switching to distinct y_i cannot make the inequality worse. This is because Hoeffding's inequality still holds when sampling is performed without replacement; in fact even better bounds are possible [11].

We now look at the imaginary part $\frac{1}{N}\sum_{i=1}^{\ell} Y'_i$ where $Y'_i = \sin(2\pi(z-z')y_i/N)$. By identical logic, we have that

$$\Pr\left[\left|\sum_{i=1}^{\ell} Y'_i\right| \geq t\right] \leq 2e^{-t^2/4\ell}$$

Combining the two inequalities with the fact that the real and imaginary parts are orthogonal gives $\Pr[|M^S_{z,z'}| \geq \sqrt{2}t/N] \leq 4e^{-t^2/4\ell}$. Setting $t = \epsilon N/\sqrt{2}$ gives that

$$\Pr\left[\left|M^S_{z,z'}\right| \geq \epsilon\right] \leq 4e^{-\epsilon^2 N^2/8\ell}$$

Taking a union bound over all $\binom{N}{2}$ off-diagonal terms[1] we obtain the lemma. ◀

▶ **Lemma 22.** *Fix subsets S, T with $|S| = \ell$. Then*

$$\tau^S_T = \frac{1}{\det(\mathbf{I} + \frac{N}{\ell}\mathbf{M}^T_S)} = \frac{1}{\det(\mathbf{I} + \frac{N}{\ell}\mathbf{M}^S_T)}$$

Proof. Recall that U' is sampled by first sampling the state $|\psi\rangle$ with support on S where each coefficient α_y is Gaussian distributed from $\mathcal{N}^{\mathbb{C}}(0,\sigma)$. Then each z is excluded from U' with probability $e^{-|\beta_z|^2}$, where $\beta_z/\sqrt{N}$ is the Fourier coefficient of z in $|\psi\rangle$. Thus, the probability that $T \cap U' = \emptyset$, conditioned on $|\psi\rangle$ is just

$$\Pr[T \cap U' = \emptyset |\ |\psi\rangle] = e^{-\sum_{z\in T}|\beta_z|^2} = e^{-N\langle\psi|\mathsf{QFT}^\dagger \Pi_T \mathsf{QFT}|\psi\rangle} = e^{-N\langle\psi|\mathbf{M}^T|\psi\rangle}$$

Now let $\mathbf{v}$ be the vector of the α_y as y varies in S. Then $|\psi\rangle$ is just $\mathbf{v}$ with zeros inserted in each position outside of S. Then we can write

$$\Pr[T \cap U' = \emptyset |\ |\psi\rangle] = e^{-N\mathbf{v}^\dagger \mathbf{M}^T_S \mathbf{v}}$$

[1] Since M^S is Hermitian, we only need to bound, say, the terms above the diagonal.

We now average over the choice of $\mathbf{v}$, which is simply distributed as $\mathcal{N}^{\mathbb{C}}(0,\sigma)^\ell$. This gives us

$$\begin{aligned}\tau_T^S &= \int_{\mathbb{C}^\ell} \Pr[\mathbf{v}] e^{-N\mathbf{v}^\dagger \mathbf{M}_S^T \mathbf{v}} d\mathbf{v} \\ &= \frac{1}{(\pi\sigma^2)^\ell} \int_{\mathbb{C}^\ell} e^{-\frac{|\mathbf{v}|^2}{\sigma^2} - N\mathbf{v}^\dagger \mathbf{M}_S^T \mathbf{v}} d\mathbf{v} \\ &= \frac{1}{(\pi\sigma^2)^\ell} \int_{\mathbb{C}^\ell} e^{-\mathbf{v}^\dagger(\sigma^{-2}\mathbf{I} + N\mathbf{M}_S^T)\mathbf{v}} d\mathbf{v} \\ &= \frac{1}{(\sigma^2)^\ell \det(\sigma^{-2}\mathbf{I} + N\mathbf{M}_S^T)} = \frac{1}{\det(\mathbf{I} + N\sigma^2\mathbf{M}_S^T)} = \frac{1}{\det(\mathbf{I} + \frac{N}{\ell}\mathbf{M}_S^T)}\end{aligned}$$

where in the last inequality we plugged in $\sigma = \sqrt{1/\ell}$. This gives us the first part of Lemma 22. Now, we observe that $\mathbf{M}_S^T = A^\dagger A$, where A is the $|T| \times \ell$ sub-matrix of QFT restricted to column indices in S and row indices in T. Then by the Weinstein–Aronszajn identity, $\det(\mathbf{I} + \frac{N}{\ell}\mathbf{M}_S^T) = \det(\mathbf{I} + \frac{N}{\ell}A^\dagger A) = \det(\mathbf{I} + \frac{N}{\ell}AA^\dagger) = \det(\mathbf{I} + \frac{N}{\ell}\mathbf{M}_T^S)$. This gives the second part of Lemma 22. ◀

Going forward, we will use $\overline{\tau}_T^S$ as shorthand for $\det(\mathbf{I} + \frac{N}{\ell}\mathbf{M}_T^S)$. Lemma 22 says that $\tau_T^S = (\overline{\tau}_T^S)^{-1}$.

We can now prove Lemmas 13, 14, 15, and 16.

4.1 Proof of Lemma 13

▶ **Lemma 13.** *For any subsets S, T, $1 + |T| \le \overline{\tau}_T^S \le 2^{|T|}$*

Proof. We first observe that the diagonal entries in $\mathbf{I} + \frac{N}{\ell}\mathbf{M}^S$ are exactly 2. Indeed, the zth diagonal entry is $1 + (N/\ell)\left(\frac{1}{N}\sum_{x\in S} e^{i2\pi zx}e^{-i2\pi zx}\right) = 2$. Moreover, $\frac{N}{\ell}\mathbf{M}^S$ and hence $\mathbf{I} + \frac{N}{\ell}\mathbf{M}^S$ is PSD. Since $\mathbf{I} + \frac{N}{\ell}\mathbf{M}_T^S$ is just a principle minor of $\mathbf{I} + \frac{N}{\ell}\mathbf{M}^S$, it is also PSD with diagonal entries also equal to 2. The determinant is then bounded by the product of the diagonal entries, giving the upper bound.

For the lower bound, we have that the eigenvalues of $\mathbf{I} + \frac{N}{\ell}\mathbf{M}_T^S$ sum to $2|T|$, and each of the $|T|$ eigenvalues is at least 1. The determinant is the product of the eigenvalues, which is minimized when one of the eigenvalues is $|T| + 1$ and the remaining $|T| - 1$ eigenvalues are 1. This gives the lower bound. ◀

4.2 Proof of Lemma 14

▶ **Lemma 14.** *For any $\epsilon > 0$, except with probability at most $2N^2 e^{-\epsilon^2\ell/2}$ over the choice of S, for all subsets T, $\overline{\tau}_T^S \ge 2^{|T|}(1 - |T|^2\epsilon)$. In this event, as long as $|T|^2\epsilon \le 1/2$, we can bound $\tau_T^S \le 2^{-|T|}(1 + 2|T|^2\epsilon)$.*

Proof. Assume all the off-diagonal entries of M^S are bounded by $2\epsilon\ell/N$, which by Lemma 21 holds except with probability $2N^2e^{-\epsilon^2\ell/2}$. We have already seen that the diagonal entries of $\mathbf{I} + \frac{N}{\ell}\mathbf{M}_T^S$ are exactly 2. The off-diagonal entries of $\mathbf{I} + \frac{N}{\ell}\mathbf{M}_T^S$ are off-diagonal entries from $\mathbf{M}^S$ but scaled by N/ℓ, and are therefore bounded by 2ϵ. By Gershgorin's circle theorem, the eigenvalues are therefore lower-bounded by $2(1 - |T|\epsilon)$. The determinant is therefore lower-bounded by this quantity raised to the $|T|$. We obtain the lemma by bounding $(1 - |T|\epsilon)^{|T|} \ge (1 - |T|^2\epsilon)$. ◀

4.3 Proof of Lemma 15

▶ **Lemma 15.** *Fix any set S of size ℓ and $\epsilon \in (0,1)$. Then except with probability $2e^{-\epsilon^2\ell/8}$, $|\ |\psi\rangle\ |^2 \in [1-\epsilon, 1+\epsilon]$, where $|\psi\rangle$ is generated as in $\mathcal{D}'_S$.*

Proof. First, $\mathbb{E}[|\ |\psi\rangle\ |^2] = \sum_{y\in S}\mathbb{E}[|\alpha_y|^2]$. Since $\alpha_y \leftarrow \mathcal{N}^{\mathcal{C}}(0, 1/\sqrt{\ell})$, the real and imaginary parts are mean-0 normal variables with variance $1/2\ell$. $|\alpha_y|^2$ is therefore distributed as the Chi-squared distribution with two degrees of freedom, scaled by $1/2\ell$, which therefore has expectation $1/\ell$. Summing over all $y \in S$ gives $\mathbb{E}[|\ |\psi\rangle\ |^2] = 1$. We also see that $|\ |\psi\rangle\ |^2$ is distributed as a Chi-squared distribution with 2ℓ degrees of freedom, scaled by $1/2\ell$. Via concentration inequalities for Chi-squared, we have that

$$\Pr\left[\left|\ |\ |\psi\rangle\ |^2 - 1\ \right| \geq 4\sqrt{x/2\ell}\right] \leq 2e^{-x}$$

Setting $\epsilon = 4\sqrt{x/2\ell}$ gives the lemma. ◀

4.4 Proof of Lemma 16

▶ **Lemma 16.** *Except with probability at most $3(N^{-1} + \ell^2 N^{-2})\epsilon^{-2} + 4N^2 e^{-\ell\epsilon^2/32}$ over the choice of S, U', we have $\left|\frac{\langle\hat{\psi}|\Pi_{U'}|\hat{\psi}\rangle}{|\ |\psi\rangle\ |^2} - 3/4\right| \leq \epsilon$. Recall $\Pi_{U'}$ is the projection operator $\sum_{z\in U'}|z\rangle\langle z|$.*

Proof. Fix $|\psi\rangle$. We first compute the expectation of $\langle\hat{\psi}|\Pi_{U'}|\hat{\psi}\rangle$ (as U varies) given $|\psi\rangle$. We will actually compute the complement $\langle\hat{\psi}|(\mathbf{I}-\Pi_{U'})|\hat{\psi}\rangle$

$$\begin{aligned}
\mathbb{E}[\langle\hat{\psi}|(\mathbf{I}-\Pi_{U'})|\hat{\psi}\rangle] &= \frac{1}{N}\mathbb{E}[\sum_{z\notin U'}|\beta_z|^2] \\
&= \frac{1}{N}\sum_{z\in\mathbb{Z}_N}\Pr[z\notin U']|\beta_z|^2 \\
&= \frac{1}{N}\sum_{z\in\mathbb{Z}_N}|\beta_z|^2 e^{-|\beta_z|^2}
\end{aligned}$$

Now we include the expectation as we vary $|\psi\rangle$, giving:

$$\mathbb{E}[\langle\hat{\psi}|(\mathbf{I}-\Pi_{U'})|\hat{\psi}\rangle] = \frac{1}{N}\sum_{z\in\mathbb{Z}_N}\mathbb{E}[|\beta_z|^2 e^{-|\beta_z|^2}]$$

Next, we bound $\mathbb{E}[|\beta_z|^2 e^{-|\beta_z|^2}]$. Since the distribution of α_y are invariant under phase change, we see that β_z is just the sum of ℓ iid variables distributed as $\mathcal{N}^{\mathcal{C}}(0, 1/\sqrt{\ell})$. This means each β_z is distributed as $\mathcal{N}^{\mathcal{C}}(0,1)$. Breaking out the real and imaginary parts lets us write

$$\begin{aligned}
\mathbb{E}[|\beta_z|^2 e^{-|\beta_z|^2}] &= \int_{\mathbb{C}}|\beta|^2 e^{-|\beta|^2}\times\frac{1}{\pi}e^{-|\beta|^2}d\beta \\
&= \frac{1}{\pi}\int_{\mathbb{C}}|\beta|^2 e^{-2|\beta|^2}d\beta = \frac{1}{\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x^2+y^2)e^{-2(x^2+y^2)}dxdy = \frac{1}{4}
\end{aligned}$$

Thus, we have that $\mathbb{E}[\langle\hat{\psi}|(\mathbf{I}-\Pi_{U'})|\hat{\psi}\rangle] = 1/4$. Now we bound the variance. By carrying out a similar calculation as before, we have:

$$\begin{aligned}
\mathbb{E}[\langle\hat{\psi}|(\mathbf{I}-\Pi_{U'})|\hat{\psi}\rangle^2] &= \frac{1}{N^2}\,\mathbb{E}[\left(\sum_{z\notin U'}|\beta_z|^2\right)] \\
&= \frac{1}{N^2}\sum_{z,z'\in\mathbb{Z}_N}\Pr[z,z'\notin U']|\beta_z|^2|\beta_{z'}|^2 \\
&= \frac{1}{N^2}\left(\sum_z \Pr[z\notin U']|\beta_z|^4 + \sum_{z\neq z'}\Pr[z\notin U']\Pr[z'\in U']|\beta_z|^2|\beta_{z'}|^2\right) \\
&= \frac{1}{N^2}\left(\sum_z |\beta_z|^4 e^{-|\beta_z|^2} + \sum_{z\neq z'}|\beta_z|^2|\beta_{z'}|^2 e^{-|\beta_z|^2-|\beta_{z'}|^2}\right)
\end{aligned}$$

We include the expectation as we vary $|\psi\rangle$. The expectation of $|\beta_z|^4 e^{-|\beta_z|^2}$ becomes

$$\mathbb{E}\left[|\beta_z|^4 e^{-|\beta_z|^2}\right] = \int_{\mathbb{C}}|\beta|^4 e^{-|\beta|^2}\times\frac{1}{\pi}e^{-|\beta|^2} = \frac{3}{4}$$

Meanwhile, to compute the expectation of $|\beta_z|^2|\beta_{z'}|^2 e^{-|\beta_z|^2-|\beta_{z'}|^2}$, we use that the co-variance matrix of $\beta_z,\beta_{z'}$ is exactly $\mathbf{M}^S_{\{z,z'\}}$. Thus

$$\begin{aligned}
&\mathbb{E}\left[|\beta_z|^2|\beta_{z'}|^2 e^{-|\beta_z|^2-|\beta_{z'}|^2}\right] \\
&= \int_{\mathbb{C}}\int_{\mathbb{C}}\frac{|\beta|^2|\beta'|^2 e^{-|\beta|^2-|\beta'|^2}}{\pi^2\det(\mathbf{M}^S_{\{z,z'\}})^2}\times e^{-\begin{pmatrix}\beta^* & (\beta')^*\end{pmatrix}\cdot\left(\mathbf{M}^S_{\{z,z'\}}\right)^{-1}\cdot\begin{pmatrix}\beta\\ \beta'\end{pmatrix}}d\beta d\beta' \\
&= \frac{1}{\pi^2\det(\mathbf{M}^S_{\{z,z'\}})^2}\int_{\mathbb{C}}\int_{\mathbb{C}}|\beta|^2|\beta'|^2 e^{-\begin{pmatrix}\beta^* & (\beta')^*\end{pmatrix}\cdot\left[\mathbf{I}+\left(\mathbf{M}^S_{\{z,z'\}}\right)^{-1}\right]\cdot\begin{pmatrix}\beta\\ \beta'\end{pmatrix}}d\beta d\beta' \\
&= \frac{1}{\pi^2\det(\mathbf{M}^S_{\{z,z'\}})^2}\times\frac{\pi^2\mathsf{Tr}(\mathbf{I}+(\mathbf{M}^S_{\{z,z'\}})^{-1})^2}{4\det(\mathbf{I}+(\mathbf{M}^S_{\{z,z'\}})^{-1})^3} \\
&= \frac{\det(\mathbf{M}^S_{\{z,z'\}})\mathsf{Tr}(\mathbf{I}+(\mathbf{M}^S_{\{z,z'\}})^{-1})^2}{4\det(\mathbf{I}+\mathcal{M}^S_{\{z,z'\}})^3}
\end{aligned}$$

Now write $\mathbf{M}^S_{\{z,z'\}} = \begin{pmatrix}1 & \epsilon_{z,z'}\\ \epsilon_{z,z'} & 1\end{pmatrix}$. By Lemma 21, we can now bound $|\epsilon_{z,z'}|$ by $\ell/4N < 1/2$, except with probability $2N^2e^{-\ell/32}$. Then we have that

$$\mathbb{E}[|\beta_z|^2|\beta_{z'}|^2 e^{-|\beta_z|^2-|\beta_{z'}|^2}] = \frac{\left(2-|\epsilon_{z,z'}|^2\right)^2}{\left(1-|\epsilon_{z,z'}|^2\right)\left(4-|\epsilon_{z,z'}|^2\right)^3} \leq \frac{1}{16}(1+|\epsilon_{z,z'}|^2)$$

where the last inequality used that $|\epsilon_{z,z'}|\leq 1/2$.

We therefore have that $\mathbb{E}[\langle\hat{\psi}|(\mathbf{I}-\Pi_{U'})|\hat{\psi}\rangle^2] \leq \frac{3}{4N}+\frac{1}{16}+\frac{1}{16N^2}\sum_{z\neq z'}|\epsilon_{z,z'}|$. Since the mean is $1/4$, the variance is therefore at most $\frac{3}{4N}+\frac{1}{16N^2}\sum_{z\neq z'}|\epsilon_{z,z'}|^2 \leq \frac{3}{4N}+\frac{1}{16N^2}\sum_{z\neq z'}\left(\frac{\ell}{4N}\right)^2 \leq \frac{3}{4N}+\frac{\ell^2}{256N^2}$. We now apply Chebyshev's inequality, to get that

$$\Pr[|\langle\hat{\psi}|(\mathbf{I}-\Pi_{U'})|\hat{\psi}\rangle - 1/4| \geq \epsilon/2] \leq \frac{\frac{3}{4N}+\frac{\ell^2}{64N^2}}{(\epsilon/2)^2} + 2N^2e^{-\ell/32}$$

Now we have that

$$\begin{aligned}
\Pr\left[\left|\frac{\langle\hat{\psi}|\Pi_{U'}|\hat{\psi}\rangle}{|\,|\psi\rangle\,|^2} - 3/4\right| \geq \epsilon\right] &= \Pr\left[\left|\frac{\langle\hat{\psi}|(\mathbf{I}-\Pi_{U'})|\hat{\psi}\rangle}{|\,|\psi\rangle\,|^2} - 1/4\right| \geq \epsilon\right] \\
&= \Pr\left[\langle\hat{\psi}|(\mathbf{I}-\Pi_{U'})|\hat{\psi}\rangle \notin |\,|\psi\rangle\,|^2 \times \left[\frac{1}{4}-\epsilon, \frac{1}{4}+\epsilon\right]\right] \\
&\leq \Pr\left[\langle\hat{\psi}|(\mathbf{I}-\Pi_{U'})|\hat{\psi}\rangle \notin \left[\frac{1}{4}-\epsilon/2, \frac{1}{4}+\epsilon/2\right]\right] \\
&\quad + \Pr[|\,|\psi\rangle\,|^2 \notin [1-\epsilon/2, 1+\epsilon/2]] \\
&\leq \left(\frac{\frac{3}{4N}+\frac{\ell^2}{64N^2}}{(\epsilon/2)^2} + 2N^2 e^{-\ell/32}\right) + 2e^{-\ell\epsilon^2/32} \\
&\leq 3(N^{-1}+\ell^2 N^{-2})\epsilon^{-2} + 4N^2 e^{-\ell\epsilon^2/32}
\end{aligned}$$

◀

4.5 Proof of Lemma 18

▶ **Lemma 18.** *If U' is sampled from a k'-wise uniform independent function and S' is potentially correlated to U' but has size at most v, then except with probability $2N^2\left(\frac{\sqrt{ek'}}{\epsilon\sqrt{N}}\right)^{k'}$ over the choice of U', S', it holds that, for any normalized state $|\phi\rangle$ with support on S', $\langle\hat{\phi}|\Pi_{U'}|\hat{\phi}\rangle \leq 1/2 + v\epsilon$, where $|\hat{\phi}\rangle = \mathsf{QFT}|\phi\rangle$.*

Proof. For a set U', let $\mathbf{M}^{U'} = \mathsf{QFT}^\dagger\Pi_{U'}\mathsf{QFT}$. Then $(\mathbf{M}^{U'})_{z,z'} = \frac{1}{N}\sum_{y\in\mathbb{Z}_N} e^{i2\pi(z-z')y/N}\xi_y$, where ξ_y is the boolean value that is 1 if $y \in U'$. Then we have that $(\mathbf{M}^{U'})_{z,z} = |U'|/N$. In expectation, $|U'|/N$ is $1/2$. We now bound how far $|U'|/N$ can deviate from $1/2$. Recall that $|U'|/N = \frac{1}{N}\sum_{y\in\mathbb{Z}_N}\xi_y$. The ξ_y are not truly independent so we cannot use standard concentration inequalities such as Hoeffding. However, we can use the following somewhat standard bound (e.g. [12]) for the $[-1,1]$-weighted sum of k'-wise independent boolean random variables:

$$\Pr\left[\left|\frac{1}{N}\sum_{y\in\mathbb{Z}_N}\xi_y - \frac{1}{2}\right| \geq \epsilon/\sqrt{2}\right] \leq 2\left(\frac{\sqrt{ek'}}{\epsilon\sqrt{N}}\right)^{k'}$$

On the other hand, for $z' \neq z'$, taking the expectation over U', we see that $\mathbb{E}_{U'}[(\mathbf{M}^{U'})_{z,z}] = 0$. We now try to bound the deviation from the expectation, by bounding the real and imaginary parts separately. We see that $\Re[(\mathbf{M}^{U'})_{z,z'}] = \frac{1}{N}\sum_{y\in\mathbb{Z}_N}\cos(2\pi(z-z')y/N)\xi_y$, which is the weighted sum of boolean random variables ξ_y, where the weights are all in $[-1,1]$. Using the $[-1,1]$-weighted sum of k'-wise independent boolean random variables again, we have:

$$\Pr[\frac{1}{N}\sum_{y\in\mathbb{Z}_N}\cos(2\pi(z-z')y/N)\xi_y \geq \epsilon/\sqrt{2}] \leq 2\left(\frac{\sqrt{ek'}}{\epsilon\sqrt{N}}\right)^{k'}$$

Combining with an identical bound on the imaginary part of $(\mathbf{M}^U)_{z,z'}$, we have that

$$\Pr[|(\mathbf{M}^{U'})_{z,z'}| \geq \epsilon] \leq 4\left(\frac{\sqrt{ek'}}{\epsilon\sqrt{N}}\right)^{k'}$$

By union-bounding over all entries of $\mathbf{M}^U$ (which only needs to count entries on the diagonal and above since $\mathbf{M}$ is Hermitian), we have except for probability at most $2N^2\left(\frac{\sqrt{ek'}}{\epsilon\sqrt{N}}\right)^{k'}$, both (1) for all z that $(\mathbf{M}^U)_{z,z} \in [1/2-\epsilon, 1/2+\epsilon]$ and (2) for all $z \neq z'$ that $|(\mathbf{M}^U)_{z,z'}| \leq \epsilon$.

Now suppose (1) and (2) hold. Now consider a supposed set S' of size v. Let $\mathbf{M}_{S'}^{U'}$ be the $v \times v$ sub-matrix whose row and column indices are in S'. Then by the Gershgorin circle theorem, all eigenvalues of $\mathbf{M}_{S'}^{U'}$ are bounded from above by $1/2 + v\epsilon$. ◀

4.6 Proof of Theorem 5

▶ **Theorem 5.** *There exists a classical oracle $\mathcal{O}$ such that* $\mathsf{QCMA}^{\mathcal{O}} \neq \mathsf{QMA}^{\mathcal{O}}$ *if and only if* $\mathsf{OI\text{-}QCMA} \neq \mathsf{OI\text{-}QMA}$

Proof. In one direction, assume a classical oracle $\mathcal{O}$ such that $\mathsf{QCMA}^{\mathcal{O}} \neq \mathsf{QMA}^{\mathcal{O}}$. Let $\mathsf{L}^{\mathcal{O}}$ be the separating language, and V a verifier for $\mathsf{L}^{\mathcal{O}}$.

We construct a language $\mathsf{OI\text{-}L}$ as follows. Let $Q(n)$ be a (polynomial) upper bound on the length of queries that V makes to $\mathcal{O}$ when given an instance x of length n. Let $\mathcal{O}_n$ be the portion of $\mathcal{O}$ corresponding to queries of length at most $Q(n)$. Then $|\mathcal{O}_n| \leq 2^{n^{\Theta(1)}}$.

Let $\mathcal{U}_n$ consist of strings of the form $(x, \mathcal{O}_n)$ where x has size n, and let $\mathsf{OI\text{-}L}$ be the set of $(x, \mathcal{O}_n)$ where $x \in \mathsf{L}^{\mathcal{O}}$. We can decide membership in $\mathsf{OI\text{-}L}$ as follows: given a witness state $|\psi\rangle$, first recover x by making n queries to $(x, \mathcal{O}_n)$. Then run $V^{\mathcal{O}_n}(x, |\psi\rangle)$. By our choice of $\mathcal{O}_n$ containing all of $\mathcal{O}$ that gets queried by V, we therefore have that $V^{\mathcal{O}_n}(x, |\psi\rangle)$ is identical to $V^{\mathcal{O}}(x, |\psi\rangle)$. As such, if V correctly decides if $x \in \mathsf{L}^{\mathcal{O}}$, then our verifier correctly decides in $(x, \mathcal{O}_n) \in \mathsf{OI\text{-}L}$. Thus, $\mathsf{OI\text{-}L} \in \mathsf{OI\text{-}QMA}$.

On the other hand, suppose there is a $\mathsf{OI\text{-}QCMA}$ verifier V_* that decides $\mathsf{OI\text{-}L}$. We can then readily obtain a QCMA verifier for $\mathsf{L}^{\mathcal{O}}$. On input instance x and witness w, simulate $V_*^{(x,\mathcal{O}_n)}(w)$ by answering queries to x with the bits of x, and queries to $\mathcal{O}_n$ by forwarding the queries to $\mathcal{O}$. If V_* decides membership in $\mathsf{OI\text{-}L}$, this exactly decides if $x \in \mathsf{L}^{\mathcal{O}}$.

We now turn to the other direction in the statement of Theorem 5. Assume that $\mathsf{OI\text{-}QCMA} \neq \mathsf{OI\text{-}QMA}$. Let $\mathcal{U}$ be the universe and $\mathsf{OI\text{-}L} \subseteq \mathcal{U}$ be the separating language, and V the $\mathsf{OI\text{-}QMA}$ verifier for $\mathsf{OI\text{-}L}$.

We construct our oracle $\mathcal{O}$ and associated language $L^{\mathcal{O}}$ as follows. $\mathcal{O}$ will be interpreted as a countably-infinite family of oracles $(\mathcal{X}_{n_i})_{i \in \mathbb{Z}^+}$ for integers $n_i \in \mathbb{Z}^+$. Let $\mathcal{O}_j = (\mathcal{X}_{n_i})_{i \leq j}$. By padding the witness with 0's size appropriately, we can assume that any potential QCMA verifier runs in quadratic time. Let $T_1, T_2, \cdots$ be an enumeration over all oracle-aided quadratic-time quantum algorithms. $\mathcal{O}$ and $\mathsf{L}^{\mathcal{O}}$ will be constructed by constructing $\mathcal{X}_{n_i}$ for $i = 1, 2, \cdots$ as follows: Consider the $\mathsf{OI\text{-}QCMA}$ verifier $V_i^{\mathcal{X}}(w)$ which has $\mathcal{O}_{i-1}$ hard-coded, and runs $T_i^{\mathcal{O}_{i-1},\mathcal{X}}(w)$. Since $\mathcal{O}_{i-1}$ is constant-sized, $V_i^{\mathcal{X}}(w)$ runs in quadratic time (though with a large constant overhead coming from $\mathcal{O}_{i-1}$). Let n_i be an integer and $\mathcal{X}_{n_i} \in \mathcal{U}_{n_i}$ be an instance such that both:

- V_i incorrectly decides $\mathcal{X}_{n_i}$ given classical witnesses/inputs. That is, either $\mathcal{X}_{n_i} \in \mathsf{OI\text{-}L}$ and $V_i^{\mathcal{X}_{n_i}}(w)$ rejects for all w, or $\mathcal{X}_{n_i} \notin \mathsf{OI\text{-}L}$ and there exists a w such that $V_i^{\mathcal{X}_{n_i}}(w)$ accepts.
- n_i is large enough so that the inputs to the function $\mathcal{X}_{n_i}$ are longer than the running time of $T_j^{\mathcal{O}_j}$ on inputs of length n_j for all $j < i$, meaning $T_j^{\mathcal{O}_j}$, and hence V_j, on inputs of length $n_j, j < i$ never query any input that is an input to $\mathcal{X}_{n_i}$.

Such an $\mathcal{X}_{n_i}$ is guaranteed to exist: that *some* $n_i, \mathcal{X}_{n_i}$ exist satisfying the first criteria follows immediately from the fact that $\mathsf{OI\text{-}L} \notin \mathsf{OI\text{-}QCMA}$. Suppose that there is no $n_i, \mathcal{X}_{n_i}$ satisfying the second criteria. This means there are only a finite number of $\mathcal{X}$ where V_i fails to correctly decide. But by hard-coding these bad instances along with the correct solutions, we can obtain a new verifier V_i' which correctly decides $\mathsf{OI\text{-}L}$, contradicting that $\mathsf{OI\text{-}L} \notin \mathsf{QCMA}$.

We then let $\mathsf{L}^{\mathcal{O}}$ consist of the strings 0^{n_i} such that $\mathcal{X}_{n_i} \in \mathsf{OI\text{-}L}$. We immediately see that $\mathsf{L}^{\mathcal{O}} \in \mathsf{QMA}^{\mathcal{O}}$: by making appropriate queries to $\mathcal{O}$, we can simulate the $\mathsf{OI\text{-}QMA}$ verifier $V^{\mathcal{X}_{n_i}}$, thereby deciding membership of 0^{n_i} in $\mathsf{OI\text{-}L}$.

We now show that $\mathsf{L}^{\mathcal{O}} \notin \mathsf{QCMA}^{\mathcal{O}}$. Consider a supposed QCMA verifier V_*. This corresponds to some T_i according to our enumeration. Then we argue that V_* incorrectly decides membership for 0^{n_i}. Indeed, we know that $T_i^{\mathcal{O}}$ in instance 0^{n_i} never queries on $\mathcal{X}_{n_j}$ for $j > i$, by our choice of n_j. Hence, it can be simulated as $T_i^{\mathcal{O}_i}$, or equivalently $T_i^{\mathcal{O}_{i-1},\mathcal{X}_{n_i}}$. Thus, V_* corresponds exactly to the verifier V_i. But we know that V_i incorrectly decides membership for $\mathcal{X}_{n_i}$. Hence, V_* is not a valid QCMA verifier. ◀

References

1. Scott Aaronson and Greg Kuperberg. Quantum versus classical proofs and advice. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*, pages 115–128, 2007. doi:10.1109/CCC.2007.27.
2. Dorit Aharonov and Tomer Naveh. Quantum NP – A survey, 2002.
3. Shalev Ben-David and Srijita Kundu. Oracle separation of QMA and QCMA with bounded adaptivity, 2024.
4. Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, October 1997. doi:10.1137/S0097539796300933.
5. Bill Fefferman and Shelby Kimmel. Quantum vs. classical proofs and subset verification. In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*, 2018. doi:10.4230/LIPIcs.MFCS.2018.22.
6. Yassine Hamoudi and Frédéric Magniez. Quantum time-space tradeoff for finding multiple collision pairs. *ACM Trans. Comput. Theory*, April 2023. doi:10.1145/3589986.
7. Xingjian Li, Qipeng Liu, Angelos Pelecanos, and Takashi Yamakawa. Classical vs quantum advice and proofs under classically-accessible oracle. In Venkatesan Guruswami, editor, *ITCS 2024*, volume 287, pages 72:1–72:19. LIPIcs, January / February 2024. doi:10.4230/LIPIcs.ITCS.2024.72.
8. Qipeng Liu. Non-uniformity and quantum advice in the quantum random oracle model. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part I*, volume 14004 of *LNCS*, pages 117–143. Springer, Cham, April 2023. doi:10.1007/978-3-031-30545-0_5.
9. Andrew Lutomirski. Component mixers and a hardness result for counterfeiting quantum money, 2011.
10. Anand Natarajan and Chinmay Nirkhe. A distribution testing oracle separating qma and qcma. In *Proceedings of the Conference on Proceedings of the 38th Computational Complexity Conference*, CCC '23, Dagstuhl, DEU, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2023.22.
11. R. J. Serfling. Probability Inequalities for the Sum in Sampling without Replacement. *The Annals of Statistics*, 2(1):39–48, 1974. doi:10.1214/aos/1176342611.
12. Terry Tao. 254a, notes 1: Concentration of measure. https://terrytao.wordpress.com/2010/01/03/254a-notes-1-concentration-of-measure/, 2010.
13. Takashi Yamakawa and Mark Zhandry. Verifiable quantum advantage without structure. In *63rd FOCS*, pages 69–74. IEEE Computer Society Press, October / November 2022. doi:10.1109/FOCS54457.2022.00014.
14. Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 758–775. Springer, Berlin, Heidelberg, August 2012. doi:10.1007/978-3-642-32009-5_44.

Formulations and Constructions of Remote State Preparation with Verifiability, with Applications

Jiayu Zhang ✉
Zhongguancun Laboratory, Beijing, China

Abstract

Remote state preparation with verifiability (RSPV) is an important quantum cryptographic primitive [20, 41]. In this primitive, a client would like to prepare a quantum state (sampled or chosen from a state family) on the server side, such that ideally the client knows its full description, while the server holds and only holds the state itself. In this work we make several contributions on its formulations, constructions and applications. In more detail:

- We first work on the definitions and abstract properties of the RSPV problem. We select and compare different variants of definitions [8, 20, 41, 19], and study their basic properties (like composability and amplification).
- We also study a closely related question of how to certify the server's operations (instead of solely the states). We introduce a new notion named *remote operator application with verifiability* (ROAV). We compare this notion with related existing definitions [38, 28, 24, 29, 31], study its abstract properties and leave its concrete constructions for further works.
- Building on the abstract properties and existing results [11], we construct a series of new RSPV protocols. Our constructions not only simplify existing results [20] but also cover new state families, for example, states in the form of $\frac{1}{\sqrt{2}}(|0\rangle |x_0\rangle + |1\rangle |x_1\rangle)$. All these constructions rely only on the existence of weak NTCF [12, 5], without additional requirements like the adaptive hardcore bit property [10, 5].
- As a further application, we show that the classical verification of quantum computations (CVQC) problem [2, 27] could be constructed from assumptions on group actions [4]. This is achieved by combining our results on RSPV with group-action-based instantiation of weak NTCF [5], and then with the quantum-gadget-assisted quantum verification protocol [17].

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases Quantum Cryptography, Remote State Preparation, Self-testing, Verification of Quantum Computations

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.96

Related Version *Full Version*: `https://arxiv.org/abs/2310.05246`

Funding *Jiayu Zhang*: This work is supported by different fundings at different time during its preparation:

- Partially supported by the IQIM, an NSF Physics Frontiers Center (NSF Grant PHY-1125565) with support of the Gordon and Betty Moore Foundation (GBMF-12500028).
- This work is partially done when the author was visiting Simons Institute for Theory of Computing.
- This work is partially done in Zhongguancun Laboratory.

Acknowledgements I would like to thank Thomas Vidick, Zhengfeng Ji, Anne Broadbent and Qipeng Liu for discussions.



© Jiayu Zhang;
licensed under Creative Commons License CC-BY 4.0



1 Introduction

1.1 Background

Development of quantum computers [6, 42, 9] leads to demands of various quantum cryptographic protocols. In a typical setting, there are a client and a remote quantum server. The client would like to achieve some quantum tasks, but it does not trust the server; thus the client would like to make use of cryptography to achieve its goal. Famous examples include quantum computation verification [27, 41], multiparty quantum computations [7], etc. In this work, we are interested in a basic and very important primitive called *remote state preparation* (RSP) [8], which we introduce below.

1.1.1 Remote state preparation: an overview

In the RSP problem, ideally, the client would like to prepare a quantum state (sampled or chosen from a state family) on the server side; thus in the end the client knows the description of the state, while the server simply holds the state. The trivial solution is to simply send the quantum state through a quantum channel. RSP asks: how could we achieve this task using cheaper resources (for example, only classical communication), possibly under computational assumptions?

Studies of RSP have a long history [32, 8]. One setting of RSP is the fully honest setting [8]. In this work, we are interested in the cryptographic setting where the server could be malicious. Then a formulation of RSP should at least have a correctness requirement and a security requirement.

The natural correctness requirement for RSP says that when the server is honest, the server gets the state while the client gets the state description. For security, there are different security notions, including blindness (or privacy, secrecy) and verifiability (or soundness) [15, 20, 40]. In this paper we focus on RSP with verifiability (RSPV). In RSPV, intuitively, the client is able to verify that in the case of passing (or called acceptance, non-aborting) the server (approximately) really gets the state, as if it is sent through a quantum channel. A malicious server who attempts to get other information by deviating from the protocol would be caught cheating by the client.

As a natural quantum task, the RSPV problem is interesting on its own. What's more, it has become an important building block in many other quantum cryptographic protocols. For example, [20] first constructs a classical channel cryptography-based RSPV and uses it to achieve classical verification of quantum computations; [19] explores more applications of RSPV; [41] takes the RSPV approach to achieve classical verification of quantum computations with linear total time complexity. Many quantum cryptographic protocols rely on the quantum channel and quantum communication, and an RSPV protocol could serve as a compiler: it allows us to replace these quantum communication steps by other cheaper resources, like the classical communication.

On the one hand, there have been many important and impressive results in this direction; on the other hand, there are also various limitations or subtleties in existing works, including formulations and constructions. Below we discuss existing works in more detail and motivate our results.

1.2 Existing Works and Motivating Questions

1.2.1 Formulations and abstract properties of RSPV

We first note that there are many variants of definitions for RSPV. For example, there are two subtlely different types of security notions, the *rigidity-based* (or isometry-based) soundness [15, 19] and *simulation-based* soundness [8, 20, 41]. Existing works do not seem to care about the differences; we note that these differences could have impact on the abstract properties of the definitions and could affect their well-behaveness. For example, we would like RSPV to have sequential composability between independent instances: if the client and the server execute an RSPV protocol for a state family $\mathcal{F}_1$, and then execute an RSPV protocol for a state family $\mathcal{F}_2$, we would like the overall protocol to be automatically an RSPV for $\mathcal{F}_1 \otimes \mathcal{F}_2$. If such a sequential composability property holds, protocols for tensor products of states could be reduced to protocols for each simple state family.

In this background, we argue that:

It's helpful to compare variants of definitions and formalize basic abstract properties.

We would like to have a more well-behaved framework for RSPV, which will lay a solid foundation for concrete constructions.

1.2.2 Definitions of the primitive for certifying the server's operations

RSPV talks about the certification of server-side *states*. A closely related question is: how could the client certify the server-side *operations*?

To address this problem, existing works raise the notion of self-testing [38, 33, 28]. One famous scenario of self-testing is in non-local games [24, 35]. In this scenario, the verifier sends questions to two spatially-separated but entangled quantum provers (or called servers). The verifier's questions and passing conditions are specially designed so that the provers have to perform specific operations to pass. This provides a way to constrain the provers' operations through only classical interactions and spatial separation, which has become a fundamental technique in the study of non-local games.

Recently a series of works [29, 18, 11, 31] study the single-server cryptographic analog of the non-local game self-testing. [29] studies the cryptographic analog of the CHSH game, where the server needs to prepare the Bell states and perform measurements on two of its registers. [30, 18] further extend it to three-qubit and N-qubit; [25, 31] make use of QFHE [26] to formulate and address the problem, where the FHE-encrypted part takes the role of one prover and the unencrypted part takes the role of the other prover.

What's common in these existing works is that they are defining the single-server cryptographic analog of the non-local game self-testing to be a protocol where the single server is playing the roles of *both* of the two provers. In this work we are interested in another viewpoint, which is:

How could we define a single-server cryptographic analog of the non-local game self-testing, where the single server is playing the role of one *of the two provers?*

1.2.3 Existing RSPV constructions

Maybe the most natural question in the direction of RSPV is:

For what state families could we achieve RSPV?

Let's quickly review what state families have been achieved in existing works. [20] achieves RSPV for $\{|+_\theta\rangle := \frac{1}{\sqrt{2}}(|0\rangle + e^{\mathrm{i}\theta\pi/4}|1\rangle), \theta \in \{0, 1, 2 \cdots 7\}\}$; independently [15] gives a candidate RSPV construction for this state family and conjectures its security. [20] achieves RSPV for tensor products of BB84 states: $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}^{\otimes n}$. [41] achieves RSPV for tensor products of $|+_\theta\rangle$ states. [11] achieves RSPV for BB84 states.

For a broader viewpoint let's also review some famous self-testing protocols. [29] achieves cryptographic self-testing for Bell states and the corresponding X/Z measurements. [30] achieves self-testing for a 3-qubit magic state and the corresponding measurements. [18] achieves self-testing for multiple Bell pairs and measurements. [31] achieves self-testing for "all-X" and "all-Z" operations.

As a summary, one significant limitations of existing works is that they could only handle simple tensor product states and operators. We consider this situation very undesirable since ideally we want a protocol that is sufficiently powerful to cover all the computationally-efficient state families. The lack of concrete protocols also restricts the applications of RSPV as a protocol compiler: suppose that we have a quantum cryptographic protocol that starts with sending states in (for example) $\{\frac{1}{\sqrt{2}}(|0\rangle |x_0\rangle + |1\rangle |x_1\rangle), x_0, x_1 \in \{0,1\}^n\}$, it's not clear how to use existing RSPV protocols to compile this quantum communication step to the classical communication. (Note that although there are mature techniques for creating this type of states with some other security notions [26, 10, 27], it's not known how to construct RSPV for it.)

Besides the existency problem, it's also desirable if we could weaken the assumptions needed or simplify existing results.

1.2.4 Classical verification of quantum computations and its assumption

We will apply our results on RSPV to the classical verification of quantum computations (CVQC) problem. In this problem a classical client would like to delegate the evaluation of a BQP circuit C to a quantum server, but it does not trust the server; thus it wants to make use of a protocol to verify the results. The first and perhaps the most famous construction is given by Mahadev [27]. Building on Mahadev's work and related results [26, 10, 27], a series of new CVQC protocols are constructed [3, 20, 14, 41, 31, 5].

However, one undesirable situation is that all the existing constructions are either based on LWE [34], or based on the random oracle heuristic, or based on some new conjecture on the reduction between security notions. In more detail:

- Most existing works [27, 14, 20] make use of NTCF with the "adaptive hardcore bit property" (or some stronger variants). The only known instantiation of this primitive from standard assumption is based on LWE.
- [5] gives an instantiation of NTCF with a weaker variant of the adaptive hardcore bit property using cryptographic group actions. However, to achieve the adaptive hardcore bit property it relies on an unproven conjecture on the reduction between these two notions. Alternatively one could make use of the random oracle to construct CVQC without the adaptive hardcore bit property [12, 41] but the instantiation of the random oracle is heuristic.
- If we only make use of NTCF without the adaptive hardcore bit property, we could do RSPV for BB84 states [11], but BB84 states are not known to be sufficient for constructing CVQC.
- [31] makes use of a quantum FHE-based approach to construct CVQC. However instantiations of quantum FHE rely on (variants of) NTCF and the classical FHE [26, 22], which still rely on the LWE assumption.

As mentioned above, a type of cryptographic assumption that is different from LWE is the cryptographic group actions, for example, supersingular isogeny [4, 13]. In this background, we ask:

Could we construct CVQC from cryptographic group actions?

1.3 Our Contributions

In this work we address or make progress to the questions above. Below we give an overview of our contributions; we refer to the full version for details.

1.3.1 Definitions and Abstract Properties of RSPV

We first work on the definitions and abstract properties of RSPV.

1. We first formulate and review the hierarchy of notions for formalizing RSPV. This includes the notions of registers, cq-states, protocols, paradigms of security definitions, etc.
2. We then formalize the notion of RSPV. We formulate the soundness as a simulation-based definition and compare this definition with several other variants (like the rigidity-based soundness, which is also popular).
3. We then study the abstract properties of RSPV including the sequential composability and amplification. This allows us to reduce the constructions of RSPV to primitives that are relatively easier to construct.

In summary, we clarify subtleties and build a well-behaved framework for RSPV, which enables us to build larger protocols from smaller or easier-to-construct components. In later part of this work we will build concrete RSPV constructions under this framework.

1.3.2 Our Cryptographic Analog of Self-testing: Remote Operator Application with Verifiability (ROAV)

We then introduce a new notion called remote operator application with verifiability (ROAV), as our answer to the question in Section 1.2.2.

Let's first review how the self-testing-based protocols typically work in the non-local game setting. Suppose the verifier would like to make use of the prover 1 and prover 2 to achieve some tasks – - for example, testing the ground state energy of a local Hamiltonian. One typical technique is to design two subprotocols π_{test} and π_{comp}. Subprotocol π_{test} is a non-local game with a self-testing property, while subprotocol π_{comp} is to test the Hamiltonian. Furthermore, the games are designed specially so that the prover 2, without communicating with the prover 1, could not decide which subprotocol the verifier is currently performing. Then the setting, the overall protocol and the security proof roughly go as follows:

- (Setting) The prover 1 and prover 2 initially hold EPR states. They receive questions from the verifier, make the corresponding measurements and send back the results.
- (Overall protocol) The verifier randomly chooses to execute either π_{test} or π_{comp} (without telling the provers the choices).
- (Security proof)
  1. To pass the overall protocol the provers have to pass π_{test} with high probability. By the property of π_{test} the operations of the prover 2 has to be close to the honest behavior.
  2. By the design of π_{test} and π_{comp}, and the fact that the prover 2 is close to the honest behavior in π_{test}, we could argue that the prover 2 is also close to the honest behavior in π_{comp}.
  3. Since we already know the prover 2 is close to be honest in π_{comp}, it's typically easy to analyze the execution of π_{comp} directly and show that it achieves the task.

Recall that we would like to define a single-server cryptographic analog of the non-local game self-testing where the single server plays the role of one of the two provers. So what does the "non-local game self-testing" mean here? Our idea is to consider both the step 1 and step 2 in the security proof template above as the "non-local game self-testing". Then let's focus on the prover 2 (as "one of the two provers" in our question) and assume the prover 1 is honest. Then we could do the following simplifications in the non-local game self-testing:

- In π_{test} we could assume the prover 1 first measures its states following the verifier's question; as a result, the joint state of the prover 1 and prover 2 becomes a cq-state where the verifer knows its description. Then the verifier also gets the measurement results from the prover 1's answer; so in the end we only need to consider the joint cq-state between the verifier and the prover 2.
- For π_{comp}, since we do not consider the step 3 above as a part of the self-testing notion, the question to the prover 1 in π_{comp} could be left undetermined. Then π_{comp} is not designed for any specific task any more, which in turn makes the self-testing a general notion.

Now the setting and the first two steps in the security proof could be updated as follows:

- (Setting) In π_{test} the client and the prover 2 initially hold a cq-state, where the verifier holds the classical part and the prover 2 holds the quantum part. In π_{comp} the prover 1 and prover 2 hold EPR states and the verifier does not have access to the prover 1's information.
- (Security proof) To make the first two steps in the security proof template above go through, we should at least require that:
  "The prover could pass in π_{test}" should imply that the prover's operation in π_{comp} is close to the desired one.

This gives us the basic intuition for formalizing our single-server cryptographic analog of the non-local game self-testing. Below we introduce the notion, which we call *reomte operator application with verifiability* (ROAV).

An ROAV for a POVM $(\mathcal{E}_1, \mathcal{E}_2, \cdots, \mathcal{E}_D)$ is defined as a tuple $(\rho_{\text{test}}, \pi_{\text{test}}, \pi_{\text{comp}})$ where:

- The setting contains the following registers. $\boldsymbol{D}$ is a client-side classical register, $\boldsymbol{Q}$ is a server-side quantum register, $\boldsymbol{P}$ is a quantum register in the environment with the same dimension with $\boldsymbol{Q}$.
- ρ_{test} is a cq-state on registers $\boldsymbol{D}$ and $\boldsymbol{Q}$.
- π_{test} is the protocol for the test mode. In the test mode ρ_{test} is used as the input state and $\boldsymbol{P}$ is empty.
- π_{comp} is the protocol for the computation mode (where the operations are applied). Denote the state of maximal entanglement (multiple EPR pairs) between $\boldsymbol{P}$ and $\boldsymbol{Q}$ as Φ. In the comp mode Φ is used as the input state and $\boldsymbol{D}$ is empty. Note that the execution of π_{comp} does not touch $\boldsymbol{P}$.

The soundness of ROAV is defined roughly as follows: for any adversary Adv, at least one of the following is true:

- In the test mode (π_{test} is executed with input state ρ_{test} against adversary Adv), the adversary gets caught cheating with significant probability;
- In the comp mode (π_{comp} is executed with input state Φ against adversary Adv), the final state gives the outcome of the following operations:
  The measurement described by $(\mathcal{E}_1, \mathcal{E}_2, \cdots, \mathcal{E}_D)$ is applied on $\boldsymbol{Q}$, and the client gets the measurement result $i \in [D]$; the corresponding output state (on register $\boldsymbol{P}$ and $\boldsymbol{Q}$) is $(\mathbb{I} \otimes \mathcal{E}_i)(\Phi)$.

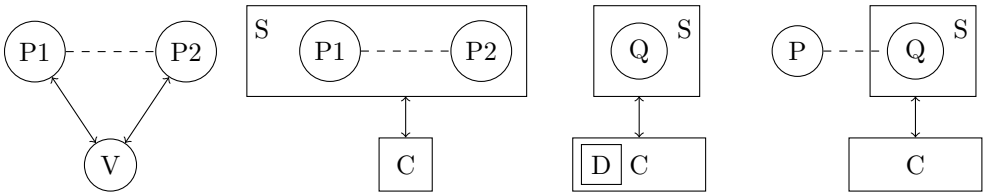


Figure 1 From the left to the right: the non-local game self-testing, previous definitions of its single party analog, and our definition (π_{test} and π_{comp} for the last two diagrams). Here $\leftrightarrow$ stands for interactions, and --- stands for quantum entanglements; C stands for the client and S stands for the server.

We finally note that the definition of ROAV also use the simulation-based security definition paradigm, like RSPV.

ROAV as the operator analog of RSPV

In the previous discussion we focus on the analog between our ROAV primitive and the non-local game self-testing. There is also another intuition that is analogous to the intuition of RSPV (see Section 1.1.1): In the ROAV problem the server is provided an undetermined input state, and the client would like to apply an operation (from an operator family) on it; in the end the client knows the description of the operation, while the server simply holds the output of the operation.

We also note that such a state-operator analog also appears in other directions like [36, 23].

Abstract constructions using ROAV

After giving the abstract definitions, we show several potential applications of our notions. We first show that ROAV is potentially a useful tool for constructing RSPV protocols for more general state families. Then we construct a Hamiltonian ground energy testing protocol based on specific RSPV and ROAV. Our construction shares similarities to Grilo's Hamiltonian verification protocol in the 2-party setting [21]. We note that these constructions are abstract constructions and concrete constructions for nontrivial ROAV remain open.

1.3.3 New RSPV Constructions

Now we introduce a series of new RSPV constructions. Our results not only give arguably simplified constructions for existing results but also cover new state families.

Overall approach

Instead of constructing each protocol directly from cryptographic assumptions, we study how different protocols could be *reduced* to existing ones in a black-box manner. Following this approach, we build a series of RSPV protocols step by step from RSPV for BB84 states $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$. Furthermore, these steps could be classified into two classes:

- In one class of steps, the reduction either has a simple intuition or is an application of the abstract properties in our framework (see Section 1.3.1) (in more detail, sequential composability and amplification).

- In the other class of steps, we could work on an "information-theoretic core" (IT-core) where the analysis is purely quantum information theoretic, and there is no appearance of computational notions like computational indistinguishability.

Our reductions are illustrated in Figure 2. We elaborate details in Section 2. As an example, let's describe the first step of our reductions. We assume an RSPV for BB84 states; then the client would like to prepare a sequence of such states in which there is only one $|+\rangle$ state and no $|-\rangle$ state. Equivalently, this could be written as $\frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle)$ where the hamming weight of $x_0 \oplus x_1$ is exactly 1. What the client needs to do is to repeat the RSPV-for-BB84 protocol for many rounds and tell the server which states it wants to keep. This could be easily understood as a repeat-and-pick process.

We argue that our approach is cleaner and easier to understand compared to the original construction in [20] at least in the following sense: In [20] the computational indistinguishability arguments and quantum information theoretic arguments are mixed together, which could lead to complicated details [1]. By separating two types of steps in the reductions explicitly, each step has a relatively simple intuition and there is much less room for complicated details.

Main results

Among these reductions, we consider the following two results particularly interesting.

▶ **Theorem 1.** *Assuming the existence of RSPV for BB84 states, there exists an RSPV for state family* $\{\frac{1}{\sqrt{2}}(|0\rangle |x_0\rangle + |1\rangle |x_1\rangle), x_0, x_1 \in \{0,1\}^n\}$.

This is an RSPV protocol for a new state family. Note that although there are plenty of works [10, 12] that allow the client to prepare these states with some other types of security (like claw-freeness, etc), as far as we know, this is the first time that an RSPV for it is constructed.

We also recover the results of RSPV for 8-basis states [20].

▶ **Theorem 2.** *Assuming the existence of RSPV for BB84 states, there exists an RSPV for state family* $\{|+_\theta\rangle := \frac{1}{\sqrt{2}}(|0\rangle + e^{\mathrm{i}\pi\theta/4}|1\rangle), \theta \in \{0,1,2\cdots 7\}\}$.

Instantiation of RSPV for BB84

We note that we still need to instantiate the RSPV for BB84 part to get a concrete protocol. Luckily the RSPV for BB84 has been studied relatively thoroughly: there are multiple constructions [19, 11] and we have a better understanding on the assumptions needed [11]. After instantiating the BB84 part by [11], we get RSPV constructions for these state families from weak NTCF, without requiring the adaptive hardcore bit property (elaborated below).

1.3.4 Application: CVQC From Cryptographic Group Actions

Now we apply our results to the classical verification of quantum computations (CVQC) problem.

As the preparation, we give a more detailed review on the variants of NTCF, and their relations to RSPV. We refer to Section 1.2.4 for a CVQC-centric background.

More backgrounds

Noisy trapdoor claw-free functions (NTCF) is a popular and powerful primitive in quantum cryptographic tasks like CVQC and RSPV. Informally, this primitive is defined to be a function family that satisfies the following requirements. Below we use f to denote a function sampled from this function family.

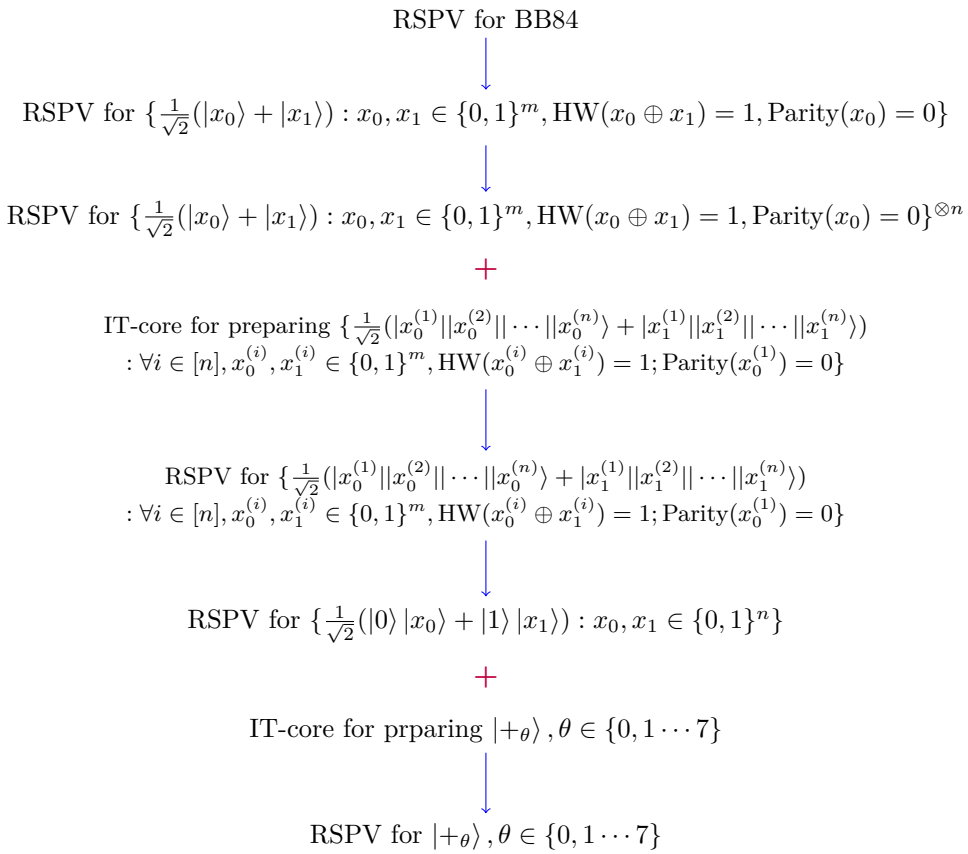


Figure 2 The reduction diagram. Here $\rightarrow$ means this step either has a simple intuition or is from the framework that we describe in Section 1.3.1; $+$ means the analysis of the IT-core below it is purely quantum information theoretic, and the RSPV above it is used to compile the IT-core to a full cryptographic protocol. HW is the hamming weight, Parity is the total parity.

- Trapdoor: this means that all the parties could evaluate f, but only the client, who holds the "trapdoor" information, could invert f.
- Noisy 2-to-1: the un-noisy 2-to-1 means that for each y in the range there exist exactly two preimages x_0, x_1 such that $f(x_0) = f(x_1) = y$. "Noisy" means that the evaluation of the function could be randomized, which makes the instantiation of the primitive easier. Below when we describe other properties we use the un-noisy version to simplify the description.
- Claw-free: the adversary could not efficiently find x_0, x_1 such that $f(x_0) = f(x_1)$.

In practice, we often use some variants of NTCF instead of the standard requirements above, to either make it more powerful or make the instantiation easier. Popular variants or additional requirements of NTCF include:

- Adaptive hardcore bit: the adversary could not find (d, y) with probability better than $\frac{1}{2}$ (the probability of random guessing) such that $d \cdot x_0 \equiv d \cdot x_1 \mod 2 \wedge d \neq 0$, where x_0, x_1 are two preimages of y.
  The RSPV for BB84 states is known to exist without this property [11], but Mahadev's constructions for CVQC [27] and RSPV for $|+_\theta\rangle$ [20] require this property.
- Extended function family: there is another function family g that is injective (instead of 2-to-1) and indistinguishable to f. Existing constructions for CVQC and RSPV for $|+_\theta\rangle$ [27, 20] also require this property.

- Inverse-polynomial correctness error (or weak correctness): this means that the 2-to-1 property is allowed to hold up to an inverse-polynomial error. This is used in [5] to define a primitive called weak TCF.

In our work we build our protocols on weak NTCF, that is, we allow inverse-polynomial correctness error and do not require additional properties like the adaptive hardcore bit property. In other words, we only use a relatively weak assumption from different variants of NTCF/TCF. This assumption could be instantiated from either LWE [10] or assumptions on group actions [5].

Applications of our results on CVQC

We first note that, [11] could be adapted easily to weak NTCF:

▶ **Theorem 3** (By [11]). *There exists an RSPV for BB84 assuming weak NTCF.*

As discussed in Section 1.3.3, this gives us a series of RSPV protocols from weak NTCF.

Then by [17], if the client samples and sends a series of $|+_\theta\rangle$ states to the server, it could do quantum computation verification using these states. Combining Fact 3, Theorem 2 and [17], we get:

▶ **Theorem 4.** *Assuming the existence of weak NTCF, there exists a classical verification of quantum computation protocol.*

Finally we recall the results in [5]:

▶ **Theorem 5** ([5]). *Under certain assumptions on cryptographic group actions, there exists a family of weak TCF.*

Thus we have:

▶ **Theorem 6.** *Under certain assumptions on cryptographic group actions, there exists a CVQC protocol.*

As an additional note, the work in [5] is largely on how to deal with the adaptive hardcore bit property; the fact that we do not need the adaptive hardcore bit may help us simplify the analysis or even construction in [5].

1.4 More Related Works

Previous versions

Compare to the previous versions, the abstract framework parts (described in Section 1.3.1, 1.3.2) of this work are sigfinicantly updated, and the concrete constructions part (described in Section 1.3.3, 1.3.4) are completely new.

RSP with other types of security

There are many works about remote state preparation but with other types of security (instead of RSPV). These works may or may not use the name "remote state preparation". As examples, [22] gives an RSP for the gadgets in [16]; [37] gives an RSP for the quantum money states. When we only consider the honest behavior, RSP is a very general notion.

Other state-operation analog

There are also several notions in quantum cryptography and complexity theory that have a state-operation analog. As examples, [36] studies the complexity of interactive synthesis of states and unitaries. [23] studies pseudorandom states and unitaries. In a sense, the relation of states and unitaries in these works is a "state-operation analog", as the RSPV and ROAV in our work.

1.5 Open Questions and Summary

Our work gives rise to a series of open questions; we consider the following two particularly interesting.

- One obvious open question coming out of this work is to give a construction for ROAV. Our work focuses on its definitions and applications in an abstract sense; an explicit construction of ROAV would allow us to instantiate these applications.
- For RSPV, although our results give new constructions for new state families, this is still far away from a general solution. Ideally we would like to have an RSPV for each computationally efficient state family. Whether this is possible and how to achieve it remain open.

In summary, two major part of this work is the framework and the constructions of RSPV protocols. By formulating and choosing notions and studying their abstract properties, we build an abstract framework for RSPV that is sufficiently well-behaved, which enables us to build more advanced, complicated protocols from more elementary, easy-to-construct components. Building on this framework, we construct a series of new RSPV protocols that not only (in a sense) simplify existing results but also cover new state families. Then we combine our results with existing works to show that the CVQC problem could be constructed from assumptions on cryptographic group actions. We also raised a new notion for certifying the server's operations. We consider our results as a solid progress in the understanding of RSPV; hopefully this will lay the foundation for further works.

2 Overview for Concrete Constructions of RSPV Protocols

We refer to the full version for the details of our contributions. In this section we elaborate our concrete RSPV constructions. As a preparation, we explain several basic notions or theorems in our framework.

- Sequential repetition: it roughly means that if protocol π_1 is an RSPV for state family $\mathcal{F}_1$, protocol π_2 is an RSPV for state family $\mathcal{F}_2$, then the sequential composition of π_1 and π_2 is an RSPV for state family $\mathcal{F}_1 \otimes \mathcal{F}_2$. This implies that, given an RSPV for some state family, its n-fold sequential composition is an RSPV for the n-fold tensor product of the state family.
- RSPV as a protocol compiler: it roughly means that, given a protocol starting with "the client prepares and sends some quantum states", we could replace this step with an RSPV protocol for these states, and the security property of the overall protocol will be preserved (approximately).
- PreRSPV and its amplification to RSPV:
  The preRSPV is defined as a pair of protocols $(\pi_{\text{test}}, \pi_{\text{comp}})$ which could be seen as a generalization of RSPV; it contains a test-mode protocol and a comp-mode protocol. The soundness is defined intuitively as follows: if π_{test} running against an adversary Adv could pass with high probability (that is, close to 1), then π_{comp} running against the same adversary Adv will satisfy the RSPV soundness.

Then a preRSPV defined above could be amplified to an RSPV by a *cut-and-choose* procedure: for each round, the client randomly chooses to execute either π_{test} or π_{comp} with some probabilities (without telling the server the choices). If the server could keep passing in many rounds, then for most of the π_{comp} rounds the target state is prepared as expected.

- PreRSPV-with-score and its amplification to RSPV:
  We also formalize a variant of preRSPV where the client counts *scores* for the server besides the pass/fail flag. The set-up is as follows: in the end of the protocol the client will record a win/lose decision in a ***score*** register, and the honest server could achieve the optimal winning probability. We use OPT to denote the optimal winning probability. (Note that when OPT = 1 the notion downgrades to the original preRSPV notion without the score.) The soundness is roughly defined as follows: if π_{test} running against an adversary Adv could win with probability close to OPT (and also pass with probability close to 1), then π_{comp} running against the same adversary Adv will satisfy the RSPV soundness.
  In the amplification procedure, the client needs to count the number of winning rounds and see whether the total score is close to the expected value of an honest execution.

Below we elaborate each step in the reduction diagram (Figure 2).

2.1 From BB84 to OneBlock and OneBlockTensor

We first build an RSPV for state family

$$\{\frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle : x_0, x_1 \in \{0,1\}^m, \text{HW}(x_0 \oplus x_1) = 1, \text{Parity}(x_0) = 0)\}$$

from the RSPV for BB84 states; this protocol is called OneBlock. The intuition is as described in Section 1.3.3: notice that states in this family could be written as sequence of BB84 states where there is only one $|+\rangle$ state and no $|-\rangle$ state; thus the client only needs to do a repeat-and-pick on RSPV-for-BB84 protocol.

Then we build the RSPV for

$$\{\frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle) : x_0, x_1 \in \{0,1\}^m, \text{HW}(x_0 \oplus x_1) = 1, \text{Parity}(x_0) = 0\}^{\otimes n} \tag{1}$$

which is the tensor products of the state family in OneBlock; this protocol is called OneBlockTensor. This is by taking the sequential repetition of the protocol OneBlock.

2.2 Construction of MultiBlock

Then we go from OneBlockTensor to construct an RSPV for the following state family:

$$\{\frac{1}{\sqrt{2}}(|x_0^{(1)}||x_0^{(2)}||\cdots||x_0^{(n)}\rangle + |x_1^{(1)}||x_1^{(2)}||\cdots||x_1^{(n)}\rangle) :$$

$$\forall i \in [n], x_0^{(i)}, x_1^{(i)} \in \{0,1\}^m, \text{HW}(x_0^{(i)} \oplus x_1^{(i)}) = 1; \text{Parity}(x_0^{(1)}) = 0\} \tag{2}$$

Roughly speaking, the client forces the server to measure the state in (1) to get (2). To see this clearly, we could first consider two blocks as an example. Suppose the server initially holds the state

$$(|x_0^{(1)}\rangle + |x_1^{(1)}\rangle) \otimes (|x_0^{(2)}\rangle + |x_1^{(2)}\rangle), \quad \forall b, i, \text{Parity}(x_b^{(i)}) = b$$

The client asks the server to measure the total parity of the strings it holds. Then if the server performs the measurement honestly, the state will collapse to:

$$\text{outcome} = 0 : |x_0^{(1)}||x_0^{(2)}\rangle + |x_1^{(1)}||x_1^{(2)}\rangle \tag{3}$$

$$\text{outcome} = 1 : |x_0^{(1)}||x_1^{(2)}\rangle + |x_1^{(1)}||x_0^{(2)}\rangle \tag{4}$$

The client could update the keys (that is, the state description) using the reported outcome and the original keys. If the client does the same for each $i = 2, 3 \cdots n$, in the honest setting the state in (1) will collapse to (2).

So what if the server cheats? One possible attack is that the server may not do the measurements and report the total parity honestly. To detect this attack, we will first construct a preRSPV (MultiBlockTest, MultiBlockComp) and use MultiBlockTest to test the server's behavior. In MultiBlockTest after getting the total parities the client will ask the server to measure all the states on the computational basis and report the measurement results; the client could check the results with its keys and see whether the results is consistent with the original keys and the total parities. As a concrete example, in (3) if the client asks the server to measure all the states, the measurement results should be either $x_0^{(1)}||x_0^{(2)}$ or $x_1^{(1)}||x_1^{(2)}$, otherwise the server is caught cheating.

After we get a preRSPV, we could amplify it to get an RSPV for (2).

Let's briefly discuss how the security proof goes through. One desirable property of the security proof of this step is that, we only need to analyze a "information-theoretic core": if we assume the initial state is (1), the analysis of the preRSPV will be purely information-theoretic, which means, the soundness holds against unbounded provers and we do not need to work on computational notions in the security analysis. After we prove the soundness of this information-theoretic part, we could prove the soundness of the overall protocol by calling the abstract properties in our framework.

2.3 Construction of KP

Then we go from MultiBlock to construct an RSPV for state family

$$\{\frac{1}{\sqrt{2}}(|0\rangle |x_0\rangle + |1\rangle |x_1\rangle) : x_0, x_1 \in \{0,1\}^n\} \tag{5}$$

This is by first calling MultiBlock to prepare a sufficiently big state in the form of (2), and then letting the client reveals suitable information to allow the server to transform (2) to (5). Let's explain the intuition.

We first note that by calculating the parity of the first block, (2) could be transformed to

$$\frac{1}{\sqrt{2}}(|0\rangle |x_0^{(1)}||x_0^{(2)}||\cdots\rangle + |1\rangle |x_1^{(1)}||x_1^{(2)}||\cdots\rangle) : \forall i, x_0^{(i)}, x_1^{(i)} \in \{0,1\}^m, \mathrm{HW}(x_0^{(i)} \oplus x_1^{(i)}) = 1 \tag{6}$$

The difference to (5) is that these keys are not sampled uniformly randomly. Note that in (6) we omit some conditions on the keys and focus on the most significant one.

Then the client will reveal lots of information about these keys, which allows the server to transform each two blocks in (6) to a pair of uniform random bits. Let's use the first two blocks as an example. The client will reveal $x_0^{(1)}$ and $x_1^{(2)}$. Then by doing xor between them and the corresponding blocks, (6) becomes:

$$\frac{1}{\sqrt{2}}(|0\rangle |0^m||000\cdots 1000\cdots||\cdots\rangle + |1\rangle |000\cdots 1000\cdots||0^m||\cdots\rangle) \tag{7}$$

Now the $000\cdots1000\cdots$ could be seen as a unary encoding of a random number in $[m]$. By choosing m to be power of 2, converting unary encoding to binary encoding and trimming out extra zeros, this state becomes

$$\frac{1}{\sqrt{2}}(|0\rangle|\gamma_0^{(1)}||\cdots\rangle + |1\rangle|\gamma_1^{(1)}||\cdots\rangle) \tag{8}$$

where $\gamma_0, \gamma_1 \in \{0,1\}^{[\log m]}$ and are uniformly random. Doing this for each two blocks in (6) gives (5).

2.4 Construction of QFac (RSPV for $|+_\theta\rangle$)

Now we have an RSPV for states $\frac{1}{\sqrt{2}}(|0\rangle|x_0\rangle + |1\rangle|x_1\rangle)$ with uniformly random x_0, x_1; we are going to construct an RSPV for the state[1]

$$|+_\theta\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{\mathrm{i}\pi\theta/4}|1\rangle), \theta \in \{0,1,2\cdots7\}.$$

We first note that existing work [20] also takes a similar approach: the client first instruct the server to prepare the state $|0\rangle|x_0\rangle + |1\rangle|x_1\rangle$, and then transforms it to $|+_\theta\rangle$.

Let's explain the constructions. The overall ideas for the construction are basically from [20] (adapted to the languages of our framework). We will first construct a preRSPV-with-score, as follows: in both the test mode and comp mode the client will instruct the server to prepare $|+_\theta\rangle$ state, then in the test mode the client will instruct the server to measure $|+_\theta\rangle$, and stores a score based on the reported result; in the comp mode $|+_\theta\rangle$ will be kept. Then once we show this protocol is indeed a preRSPV, we could amplify it to an RSPV as given in our framework.

The first step is to allow the honest server to transform $\frac{1}{\sqrt{2}}(|0\rangle|x_0\rangle + |1\rangle|x_1\rangle)$ to $|+_\theta\rangle$. There are multiple ways to do it, for example:

1. The client will first instruct the server to introduce a phase of $e^{\mathrm{i}\pi\theta_{2,3}/4}, \theta_{2,3} \in \{0,1,2,3\}$ where $\theta_{2,3}$ are hidden in the server's view. This could be done by selecting the xor of the first two bits of x_0, x_1 as θ_2, θ_3. Then on the one hand $\theta_{2,3} = 2\theta_2 + \theta_3$ will be completely hidden; on the other hand using the phase-table-like technique in [39, 41] the honest server could introduce a phase of $e^{\mathrm{i}\pi\theta_{2,3}/4}$ to the x_1 branch up to a global phase.
2. The server does a Hadamard measurement on each bit of the x-part and get a measurement result d; this introduces a phase of $e^{\mathrm{i}\pi(d\cdot(x_0+x_1))}$ to the qubits. Then the server sends back d to the client and the client could calculate $\theta_1 = (d\cdot(x_0+x_1)) \mod 2$. The server holds a single qubit in the state $|+_\theta\rangle, \theta = 4\theta_1 + 2\theta_2 + \theta_3$.

Now in the comp mode we are done. In the test mode the client will continue to ask the server to make measurement on a random basis $|+_\varphi\rangle, |+_{\varphi+4}\rangle, \varphi \leftarrow_r \{0,1,2\cdots7\}$. First we could see that when $\theta = \varphi$ the measurement will collapse to $|+_\varphi\rangle$ with probability 1, and when $\theta = \varphi + 4$ the measurement will collapse to $|+_{\varphi+4}\rangle$ with probability 1. Thus the client could record a "win" score if he has seen such an outcome. But solely doing this does not give us a full control on the server's behavior and states; an important idea is that, when φ is close to θ (or $\theta + 4$), the measurement should also collapse to $|+_\varphi\rangle$ (or $|+_{\varphi+4}\rangle$, correspondingly). This gives some probability of losing even for an honest server; however by analyzing the game it's possible to say "if the server wins with probability close to the optimal winning probability, the state before the testing measurement has to be close to the target state up to an isometry", which is still sufficient for amplification.

[1] The protocol name QFac is from [15].

Security proofs, existing works and their limitations, and our approach

So how could the quoted claim just now be proved? Existing works like [20, 15] has already done a lot of works on this part. In [15] the authors introduce a notion called *blind self-testing*. In more detail, let's denote the server-side state by the time that the comp mode is done corresponding to the client-side phase θ as ρ_θ. (In other words, the overall state is $\sum_\theta |\theta\rangle\langle\theta| \otimes \rho_\theta$.) Then the blind self-testing requires that the state $\rho_{\theta_{2,3}} + \rho_{\theta_{2,3}+4}$ is the same for any $\theta_{2,3}$, which is called (information-theoretic) *basis-blindness* (here $\theta_{2,3}$ is the "basis" and the notion means that the basis is completely hidden after randomization of θ_1). [15] shows that if the initial state satisfies the basis blindness property, the claim "high winning probability $\Rightarrow$ close to the target state up to an isometry" holds.

However, the proof of this claim given in [15] does not generalize to the computational analog of basis blindness. [15] does not solve the problem and leave the security of the whole protocol as a conjecture. [20] makes use of computational indistinguishability between states in the form of $\rho_{\theta_{2,3}} + \rho_{\theta_{2,3}+4}$ together with some quantum information theoretic arguments to prove the claim; in their proofs computational indistinguishability and quantum information theoretic arguments are mixed together, which could be complicated to work on and lead to sophisticated details [1].

Our new hammer for getting rid of this problem is the KP protocol, which is an RSPV for $|0\rangle|x_0\rangle + |1\rangle|x_1\rangle$. (Note that in previous works the preparation of $|0\rangle|x_0\rangle + |1\rangle|x_1\rangle$ is not known to satisfy the RSPV soundness in the malicious setting.) By starting from KP, we are able to prove the security in a much nicer way:

1. The "information-theoretic core": in the security proof we could first simply assume the server holds exactly the state $|0\rangle|x_0\rangle + |1\rangle|x_1\rangle$; then we could prove the ρ_θ generated from it has the (information-theoretic) basis blindness property; then the analysis of the testing on $|+_\theta\rangle$ is basically from existing results [20, 15].
2. Once we complete the analysis of this information-theoretic core, we could compile it using results in our framework to an RSPV.

2.5 A Summary

In summary we have achieve each step of the reductions as shown in Figure 2. The construction of OneBlock is by a repeat-and-pick procedure, OneBlockTensor is by the sequential composition, KP is by revealing some information to allow the server to transform the state to some other forms. For constructions of MultiBlock and QFac, we first analyze an "information-theoretic core" where we only need to work on statistical closeness, and compile the IT-core (IT=information-theoretic) to a full protocol by calling existing soundness properties.

References

1 Discussion with Thomas Vidick on details of paper "computationally secure and composable remote state preparation", 2024.

2 Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive proofs for quantum computations. In Andrew Chi-Chih Yao, editor, *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 453–469. Tsinghua University Press, 2010. URL: `http://conference.iiis.tsinghua.edu.cn/ICS2010/content/papers/35.html`.

3 Gorjan Alagic, Andrew M. Childs, Alex B. Grilo, and Shih-Han Hung. Non-interactive classical verification of quantum computation. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part III*, volume 12552 of *Lecture Notes in Computer Science*, pages 153–180. Springer, 2020. doi:10.1007/978-3-030-64381-2_6.

4 Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 411–439, Berlin, Heidelberg, 2020. Springer. doi:10.1007/978-3-030-64834-3_14.

5 Navid Alamati, Giulio Malavolta, and Ahmadreza Rahimi. Candidate trapdoor claw-free functions from group actions with applications to quantum protocols. In Eike Kiltz and Vinod Vaikuntanathan, editors, *Theory of Cryptography - 20th International Conference, TCC 2022, Chicago, IL, USA, November 7-10, 2022, Proceedings, Part I*, volume 13747 of *Lecture Notes in Computer Science*, pages 266–293, Berlin, Heidelberg, 2022. Springer. doi:10.1007/978-3-031-22318-1_10.

6 Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019. doi:10.1038/s41586-019-1666-5.

7 James Bartusek. Secure quantum computation with classical communication. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part I*, volume 13042 of *Lecture Notes in Computer Science*, pages 1–30, Cham, 2021. Springer. doi:10.1007/978-3-030-90459-3_1.

8 Charles H. Bennett, David P. DiVincenzo, Peter W. Shor, John A. Smolin, Barbara M. Terhal, and William K. Wootters. Remote state preparation. *Phys. Rev. Lett.*, 87:077902, July 2001. doi:10.1103/PhysRevLett.87.077902.

9 Dolev Bluvstein et al. A quantum processor based on coherent transport of entangled atom arrays. *Nature*, 604(7906):451–456, 2022. doi:10.1038/s41586-022-04592-6.

10 Zvika Brakerski, Paul F. Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 320–331. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00038.

11 Zvika Brakerski, Alexandru Gheorghiu, Gregory D. Kahanamoku-Meyer, Eitan Porat, and Thomas Vidick. Simple tests of quantumness also certify qubits. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part V*, volume 14085 of *Lecture Notes in Computer Science*, pages 162–191. Springer, 2023. doi:10.1007/978-3-031-38554-4_6.

12 Zvika Brakerski, Venkata Koppula, Umesh V. Vazirani, and Thomas Vidick. Simpler proofs of quantumness. In Steven T. Flammia, editor, *15th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2020, June 9-12, 2020, Riga, Latvia*, volume 158 of *LIPIcs*, pages 8:1–8:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, May 2020. doi:10.4230/LIPIcs.TQC.2020.8.

13 Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 395–427, Berlin, Heidelberg, 2018. Springer. doi:10.1007/978-3-030-03332-3_15.

14 Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. Classical verification of quantum computations with efficient verifier. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part III*, volume 12552 of *Lecture Notes in Computer Science*, pages 181–206, Cham, 2020. Springer. doi:10.1007/978-3-030-64381-2_7.

15 Alexandru Cojocaru, Léo Colisson, Elham Kashefi, and Petros Wallden. Qfactory: Classically-instructed remote secret qubits preparation. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I*, volume 11921 of *Lecture Notes in Computer Science*, pages 615–645. Springer, 2019. doi:10.1007/978-3-030-34578-5_22.

16 Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum homomorphic encryption for polynomial-sized circuits. *IACR Cryptol. ePrint Arch.*, 2016:559, 2016. URL: http://eprint.iacr.org/2016/559.

17 Samuele Ferracin, Theodoros Kapourniotis, and Animesh Datta. Reducing resources for verification of quantum computations. *Phys. Rev. A*, 98:022323, August 2018. doi:10.1103/PhysRevA.98.022323.

18 Honghao Fu, Daochen Wang, and Qi Zhao. Parallel self-testing of EPR pairs under computational assumptions. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*, volume 261 of *LIPIcs*, pages 64:1–64:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ICALP.2023.64.

19 Alexandru Gheorghiu, Tony Metger, and Alexander Poremba. Quantum cryptography with classical communication: parallel remote state preparation for copy-protection, verification, and more. *IACR Cryptol. ePrint Arch.*, page 122, 2022. URL: https://eprint.iacr.org/2022/122.

20 Alexandru Gheorghiu and Thomas Vidick. Computationally-secure and composable remote state preparation. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 1024–1033. IEEE Computer Society, 2019. doi:10.1109/FOCS.2019.00066.

21 Alex B. Grilo. A simple protocol for verifiable delegation of quantum computation in one round. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 28:1–28:13, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2019.28.

22 Aparna Gupte and Vinod Vaikuntanathan. How to construct quantum fhe, generically. *IACR Cryptol. ePrint Arch.*, page 893, June 2024. URL: https://eprint.iacr.org/2024/893, arXiv:2406.03379.

23 Zhengfeng Ji, Yi-Kai Liu, and Fang Song. Pseudorandom quantum states. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 126–152, Cham, 2018. Springer International Publishing. doi:10.1007/978-3-319-96878-0_5.

24 Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. Mip* = RE. *Commun. ACM*, 64(11):131–138, October 2021. doi:10.1145/3485628.

25 Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Lisa Yang. Quantum advantage from any non-local game. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, STOC 2023, pages 1617–1628, New York, NY, USA, 2023. ACM. doi:10.1145/3564246.3585164.

26 Urmila Mahadev. Classical homomorphic encryption for quantum circuits. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 332–338. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00039.

27 Urmila Mahadev. Classical verification of quantum computations. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 259–267. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00033.

28 Dominic Mayers and Andrew Chi-Chih Yao. Self testing quantum apparatus. *Quantum Inf. Comput.*, 4(4):273–286, July 2004. doi:10.26421/QIC4.4-3.

29 Tony Metger and Thomas Vidick. Self-testing of a single quantum device under computational assumptions. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 19:1–19:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ITCS.2021.19.

30 Akihiro Mizutani, Yuki Takeuchi, Ryo Hiromasa, Yusuke Aikawa, and Seiichiro Tani. Computational self-testing for entangled magic states. *Phys. Rev. A*, 106(1):L010601, 2022. doi:10.1103/PhysRevA.106.L010601.

31 Anand Natarajan and Tina Zhang. Bounding the quantum value of compiled nonlocal games: From CHSH to BQP verification. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 1342–1348. IEEE, 2023. doi:10.1109/FOCS57990.2023.00081.

32 Arun Kumar Pati. Minimum cbits required to transmit a qubit. *Phys. Rev. A*, 63:014320, 2001. doi:10.1103/PhysRevA.63.014320.

33 Sandu Popescu and Daniel Rohrlich. Which states violate bell's inequality maximally? *Physics Letters A*, 169(6):411–414, 1992. doi:10.1016/0375-9601(92)90819-8.

34 Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), September 2009. doi:10.1145/1568318.1568324.

35 Ben W. Reichardt, Falk Unger, and Umesh V. Vazirani. Classical command of quantum systems. *Nat.*, 496(7446):456–460, 2013. doi:10.1038/nature12035.

36 Gregory Rosenthal and Henry Yuen. Interactive proofs for synthesizing quantum states and unitaries. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 112:1–112:4. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ITCS.2022.112.

37 Omri Shmueli. Public-key quantum money with a classical bank. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, STOC 2022, pages 790–803, New York, NY, USA, 2022. ACM. doi:10.1145/3519935.3519952.

38 Stephen J. Summers and R. Werner. Maximal Violation of Bell's Inequalities Is Generic in Quantum Field Theory. *Commun. Math. Phys.*, 110:247–259, 1987. doi:10.1007/BF01207366.

39 Jiayu Zhang. Delegating quantum computation in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 30–60, Cham, 2019. Springer. doi:10.1007/978-3-030-36033-7_2.

40 Jiayu Zhang. Succinct blind quantum computation using a random oracle. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1370–1383, New York, NY, USA, 2021. ACM. doi:10.1145/3406325.3451082.

41 Jiayu Zhang. Classical verification of quantum computations in linear time. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 46–57, Los Alamitos, CA, USA, November 2022. IEEE. doi:10.1109/FOCS54457.2022.00012.

42 Qingling Zhu, Sirui Cao, Fusheng Chen, Ming-Cheng Chen, Xiawei Chen, Tung Hsun Chung, Hui Deng, Yajie Du, Daojin Fan, Ming Gong, Cheng Guo, Chu Guo, Shaojun Guo, Lian-Chen Han, Linyin Hong, Heliang Huang, Yongheng Huo, Liping Li, Na Li, Shaowei Li, Yuan Yuan Li, Futian Liang, Chun Lin, Jin Lin, Haoran Qian, Dan Qiao, Hao Rong, Hong-Bo Su, Lihua Sun, Liangyuan Wang, Shiyu Wang, Dachao Wu, Yulin Wu, Yu Xu, Kai Yan, Weifeng Yang, Yang Yang, Yangsen Ye, Jian Hua Yin, Chong Ying, Jiale Yu, Chen Zha, Cha Zhang, Haibin Zhang, Kaili Zhang, Yiming Zhang, Han Zhao, You-Wei Zhao, Liang Zhou, Chaoyang Lu, Cheng-Zhi Peng, Xiaobo Zhu, and Jian-Wei Pan. Quantum computational advantage via 60-qubit 24-cycle random circuit sampling. *Science bulletin*, 67 3:240–245, 2021. URL: https://api.semanticscholar.org/CorpusID:237442167.